Universidad de Concepción
Dirección de Postgrado
Facultad de Ingeniería – Programa de Magíster en Ciencias de Ingeniería con Mención en Ingeniería Eléctrica

Desarrollo de método basado en Deep Learning para la Extracción de Entidades desde información clínica

Tesis para optar al grado de Magíster en Ciencias de Ingeniería con Mención en Ingeniería Eléctrica

JAIME ANDRÉS JIMÉNEZ RUIZ
CONCEPCIÓN-CHILE
2020

Profesor Guía: Rosa L. Figueroa I., Ph. D
Profesor Co-Guía: Guillermo F. Cabrera V., Ph. D

Dpto. de Ingeniería Eléctrica, Facultad de Ingeniería
Universidad de Concepción

Resumen

La información digital está aumentando día a día producto de la digitalización de diferentes servicios. En particular, en el área de la salud, la masificación de la ficha médica electrónica se convierte en una fuente de valiosa información. En particular, para los objetivos del presente trabajo, la detección de eventos de efectos adversos o efectos secundarios por medicamentos es un tipo de información que puede ser extraída desde campos de texto libre en documentos clínicos. Una detección temprana permite la toma de decisiones clínicas rápida y sistémicas, con beneficio de la salud del paciente.

El reconocimiento de entidades médicas relevantes corresponde al primer paso para la posterior extracción de eventos adversos. Esta tarea, puede ser modelada como un problema de clasificación secuencial. Desde este punto de vista del aprendizaje automático, se le ha entregado soluciones con el uso de clasificadores tradicionales y probabilísticos. El aprendizaje profundo o *Deep Learning* se ha presentado como una solución novedosa para este tipo de tareas, debido a sus buenos resultados al ser entrenado a partir de grandes conjuntos de datos. En este campo, la solución actual propuesta, es el uso de una red neuronal *Long-Short Term Memory o LSTM* (tipo de red neuronal recurrente con una celda de memoria) de tipo *bidireccional* (que recibe a la secuencia desde ambas direcciones) en conjunto a una capa de salida probabilística, donde la entrada a la red corresponde a las secuencias de palabras representadas por *word-embeddings* (que entrega una representación vectorial a la información semántica de cada palabra). En el presente trabajo se propone el desarrollo de un sistema para *la extracción automática de entidades relacionadas a efectos adversos por medicamentos en textos clínicos escritos en lenguaje natural, entrenado a partir de un corpus previamente anotado, y basado en técnicas de Deep Learning*.

Como resultados del presente trabajo de investigación: (i) se exploró el corpus a utilizar tanto el texto como sus anotaciones, (ii) se realizó pre-procesamiento (normalización, eliminación de carácteres especiales, y tokenización) de los documentos y su preparación para el entrenamiento, (iii) se replicó y entrenó el modelo propuesto por el estado-del-arte mediante el uso de Keras, logrando un F1-score de un 79% sobre las clases de interés, (iv) se re-etiquetó el corpus mediante un algoritmo de aprendizaje no supervisado de clustering (v) se re-entrenó el modelo del estado del arte utilizando las nuevas etiquetas y se evaluó el desempeño sobre las clases etiquetadas por expertos, logrando un F1-score macro de 79%, pero un F1-score inverso de 77%, con un aumento porcentual del "*Recall*" sobre las clases de menor presencia en el corpus.

"(...) Chemistry is the study of matter, but I prefer to see it as the study of change. It's growth, then decay, then transformation."
Walter White, Breaking Bad S01E01

Agradecimientos

Agradezco a mi familia y amigos, por su compañía y motivación para terminar este viaje.

Tabla de Contenidos

Lista de Tablas

Lista de Figuras

Abreviaciones

Mayúsculas

ADE : Adverse Droug Event (Evento de efecto Adverso a Medicamentos)
CBOW : Continuous Bag-of-Words (Bolsa de Palabras Continua)
CNN : Convolutional Neural Network (Red Neuronal Convolucional)
CO : Co-reference Resolution (Correferencia)
CRF : Conditional Random Fields (Campo Aleatorio Condicional)
EE : Event Extraction (Extracción de Eventos)
EHR : Electronic Health Record (Historial Clínico Electrónico)
FP : False Positives (Falsos Positivos)
FN : False Negatives (Falsos Negativos)
HMM : Hidden Markov Models (Modelo Oculto de Markov)
IE : Information Extraction (Extracción de Información)
IRB : Institutional Review Board (en referencia a Aprobación por el Comité de Ética)
IT : Information Technology (Tecnologías de Información)
LSA : Latent Semantic Analysis (Análisis Semántico Latente)
LSTM : Long Short-Term Memory
MINSAL : Ministerio de Salud (Chile)
ML : Machine Learning (Aprendizaje de Máquina)
MEMM : Maximum Entropy Markov Models (Modelos de Markov de Máxima Entropía)
NER : Named Entity Recognition (Reconocimiento de Entidades)
NLP : Natural Language Processing (Procesamiento del Lenguaje Natural)
NLTK : Natural Language Toolkit
NN : Neural Network (Red Neuronal)
POS : Part of Speech Tagging (Etiquetado gramatical)
RE : Relation Extraction (Extracción de Relaciones)
RNN : Recurrent Neural Network (Red Neuronal Recurrente)
SVD : Singular Value Descomposition (Descomposición por Valores Singulares)
TP : True Positives (Verdaderos Positivos)

Minúsculas

Bi : Bidirectional (Bidireccional)
BioNER : Biomedical NER (Reconocimiento de Entidades Biomédicas)
Pr : Precisión
Re : Recall
ReLU : Rectified Linear Unit (Unidad Lineal Rectificada)
devset : Development Test Set (Set de prueba durante entrenamiento)

Capítulo 1. Introducción

1.1. Introducción General

Nos encontramos en la época del *Big Data*, con un aumento de los dispositivos capaces de generar y almacenar datos, en conjunto a la digitalización de servicios, se espera que para 2025 la información digital almacenada a nivel mundial alcance los 175 Zettabytes, de acuerdo a una proyección publicada por la *International Data Corporation* en noviembre de 2018 [1]. En el área de la salud, en particular, el aumento de datos digitales ha sido impulsado por incentivos monetarios y legislativos de algunos gobiernos, lo que proyecta un crecimiento sostenido en el mercado global de las tecnologías de información (IT, por *Information Tehnology*) en salud hasta el año 2025 [2]. Esta tendencia a la digitalización en salud se ha visto acelerada en este último año, debido a la pandemia del Covid-19, impulsando la implementación de herramientas digitales para la telemedicina, ante la necesidad de decisiones basadas en datos apoyadas por inteligencia artificial, vigilancia de casos activos a nivel territorial y atención de pacientes a distancia para salvaguardar el distanciamiento físico 81[3]. En nuestro país, a través del Ministerio de Salud (MINSAL) y su plan de Estrategia Digital o e-Salud [4][5] se ha apoyado la digitalización del servicio, promoviendo el uso de la ficha clínica electrónica y servicios de telemedicina. El actual gobierno, considera como meta de su Agenda Digital alcanzar un 100% de uso de ficha electrónica, para el año 2020 [6].

La digitalización de la información de centros clínicos y hospitalarios no solo permite el reemplazo del papel en el almacenamiento de datos, sino que genera oportunidades de introducir herramientas de análisis de datos [7] para el aprovechamiento de los nuevos datos digitales disponibles.

La ficha clínica electrónica o EHR (de sus siglas en inglés por Electronic Health Record) es una versión digital de la ficha del paciente, contiene información en tiempo real y centrada en el paciente, sobre su historial clínico, resultados de exámenes, diagnósticos y tratamientos [8]. Un EHR está formado por campos estructurados y no estructurados. Algunas de las fuentes más valiosas de información, como las indicaciones de alta médica y notas médicas, corresponden a campos no estructurados de texto libre, por lo que, para su análisis, corresponde utilizar herramientas de procesamiento de lenguaje natural (NLP, del inglés *Natural Language Processing*) [9][10].

Para obtener información útil es necesario realizar una tarea de extracción de la misma. Formalmente, se define como extracción de información o IE (del inglés, *Information Extraction*) a

la tarea de "*identificar instancias de una entidad o clase predefinida, relaciones y eventos en textos de lenguaje natural, determinando sus propiedades o argumentos*" [11].

Las tareas clásicas de IE se dividen en: *Named Entity Recognition* (NER), *Co-reference Resolution* (CO), *Relation Extraction* (RE) y *Event Extraction* (EE) [11].

Un "*Evento de Efectos Secundarios o Efectos Adversos provocado por Medicamentos"* o ADE (del inglés, *Adverse Drug Event*), es un evento relevante que puede ser identificado en textos médicos. Un estudio realizado en Estados Unidos determinó que la mortalidad asociada a ADE varía entre 0.08-0.12 por cada 100.000 habitantes [12]. En Chile, existen casos como el ocurrido el año 2014 en el Hospital de Melipilla, donde eventos de efectos adversos no identificados llevaron a la muerte de dos pacientes [13]. Una identificación correcta y rápida, no solo es un deber ético de cualquier servicio de salud, sino que va en respuesta a la tendencia hacia un enfoque sistémico para la vigilancia de errores médicos y a los efectos adversos [14].

Un evento ADE está formado por múltiples entidades médicas, empezando por el propio *efecto adverso*, seguido por el *medicamento* que lo provocó, la *frecuencia*, *dosis* y *vía de administración*, además del *diagnóstico* para el que se recetó. Estas entidades describen información relevante que puede ser extraída de textos médicos de EHR [9][10]. Su extracción corresponde a una tarea NER (reconocimiento de entidades) el cuál es el primer paso para responder a tareas superiores como la identificación del evento, atributos y sus relaciones. Algunos acercamientos clásicos para enfrentar esta tarea contemplan una serie de heurísticas, basadas en patrones o reglas generadas por conocimiento previo (*Knowledge-Based* o *Knowledge Engineering*)[15]. Desde mediados de los años 90's, se dio un giro hacia el uso de técnicas de *Machine Learning* [11]. Bajo este acercamiento, el texto es considerado como una secuencia de tokens (unidades mínimas con significado, en este caso particular, palabras) y se utilizan métodos de etiquetado secuencial como *Hidden Markov Models* (HMMs) o *Conditional Random Fields* (CRF) [16]. En los últimos años, se ha producido un aumento en el uso de redes neuronales profundas o *Deep Learning* (DL) para múltiples áreas del conocimiento, demostrando resultados superiores a otros métodos para la tarea NER aplicada a textos biomédicos [9][10].

La extracción automática de efectos adversos se encuentra en pleno desarrollo, bajo este contexto en 2018, se realizó el challenge "*NLP Challenges for Detecting Medication and Adverse Drug Events from Electronic Health Records (MADE1.0)"*[17], donde se liberó un corpus de cerca de mil documentos de EHR, de-identificados y previamente anotados.

La presente Tesis de investigación se presenta para optar al grado de *Magíster en Ciencias de la Ingeniería con Mención en Ingeniería Eléctrica*. Se propone el uso de técnicas de aprendizaje no supervisado, para contribuir en el reconocimiento de entidades previamente anotadas con baja presencia en un corpus con alto desbalance de clases. Aplicado en el ámbito de textos médicos, previamente anotados para la detección de eventos de *Efectos Adversos a Drogas*.

1.2. Temario

(i) **Capítulo 1**: En este capítulo se introduce de forma general al tema de investigación, analizando la realidad actual y la importancia de la detección de eventos a efectos adversos. Con un breve repaso del estado del arte y la propuesta de tema.

(ii) **Capítulo 2:** Se presentan los fundamentos teóricos en el marco de la presente investigación propuesta, describiendo el proceso de extracción de información y, en particular, en el reconocimiento de entidades. Para esto se analizó la representación de textos, evaluación y fundamentos de los principales métodos utilizados para la resolución de esta tarea. Con un especial énfasis en los modelos de *Deep Learning*, su diseño y su entrenamiento.

(iii) **Capítulo 3:** Se presenta una revisión bibliográfica del estado del arte en el reconocimiento de entidades para textos biomédicos. Considerando las características usadas para la representación de los textos y la evolución de modelos de clasificación como Máquinas de soporte vectorial, a modelos probabilísticos como Cadenas de Markov y Campo Aleatorio Condicional (CRF). Se revisan los resultados documentados con el uso de *Deep Learning*, incluyendo redes convolucionales y recurrentes, además de modelos combinados con CRF.

(iv) **Capítulo 4:** En este capítulo se busca explicitar la motivación del presente trabajo de investigación. Determinando los objetivos e hipótesis propuestos, considerando el desarrollo de un modelo de clasificación de entidades médicas para la detección de efectos adversos.

(v) **Capítulo 5:** Se describe el corpus a utilizar y se detalla la metodología para dar respuesta a los objetivos propuestos. Abarcando desde el pre-procesamiento de los documentos, descripción del corpus, además del diseño, entrenamiento y evaluación de sistema de *Deep Learning.*

(vi) **Capítulo 6:** Se presentan los resultados considerando la replicación del estado-del-arte, y visualización de los resultados sobre el modelo propuesto.

(vii) **Capítulo 7:** Se realiza análisis y discusión de los resultados del trabajo actual, además, de trabajo futuro enmarcado en la línea de investigación.

Capítulo 2. Fundamentos Teóricos

2.1. Introducción

En el presente capítulo se presentan los antecedentes referentes al área de "*Extracción de Información*", enfocándose, en particular, en la tarea NER (reconocimiento de entidades previamente anotadas) aplicada a textos médicos. Se revisa, además, la representación de textos para esta tarea, algunos métodos y algoritmos para su resolución, y el actual estado del arte en relación al uso de *Deep Learning*.

2.2. Extracción de Información

Previamente, se definió como Extracción de Información o IE (del inglés, *Information Extraction*) a la tarea de "*identificar instancias de una entidad o clase predefinida, relaciones y eventos en textos de lenguaje natural, determinando sus propiedades o argumentos*" [11]. En forma general, IE engloba aquellas tareas que buscan convertir la información no estructurada presente en textos libres, en datos estructurados, por ejemplo, para poblar una base de datos relacional [18].

Las tareas clásicas en este campo son:

(i) **Named Entity Recognition (NER):** Se refiere a la tarea de reconocimiento de entidades previamente definidas. Puede ser visto como un problema de clasificación secuencial de etiquetado de texto. Ejemplos comunes son el reconocimiento de nombres de personas, lugares u organizaciones [18].

(ii) **Co-reference Resolution (CO):** Se refiere a la tarea de asociar co-referencias de entidades, es decir, asociar entidades que tengan el mismo significado. Este se puede dar por el uso de siglas o sinónimos (ej. "*General Electric*" y "*GE*"), por correferencia pronominal al reemplazar una entidad por un pronombre (ej. "*Andrés*" por "*Él*"), por correferencia nominal (ej. "*Microsoft*" por "*La compañía*") y por correferencia implícita (ej. En el texto: *"Berlusconi visitó el lugar del desastre. Sobrevoló la región por helicóptero."* la segunda oración omite al sujeto) [11].

(iii) **Relation Extraction (RE):** Se refiere a la tarea de detección y clasificación de relaciones predefinidas entre entidades. Generalmente son de tipo binaria (entre dos entidades) y en un sentido como "*hijo-de*", "*empleado-de*", "*miembro-de*", etc. Por

ejemplo, en el texto "*Steve Jobs trabaja en Apple*" implicaría una relación de tipo "*empleado-de",* donde *[Steve Jobs]→ "empleado-de":[Apple]* [11][18].

(iv) **Event Extraction (EE):** Se refiere a la identificación de eventos con sus entidades y correspondientes relaciones. Se espera responder "*Quién realizó qué, a quién, cuándo, dónde, a través de qué y por qué*" [11].

2.2.1 NER

La tarea NER es el paso inicial y el más importante para realizar cualquier tarea de Extracción de Información. En forma general, una entidad sería aquella a la que nos referimos como un *pronombre propio* (TABLA 2.1) como nombres de personas, organizaciones o lugares. Sin embargo, esta definición puede ser extendida de acuerdo al dominio de aplicación [11], por ejemplo para aplicaciones de bioinformática [19] en la detección de entidades asociadas a eventos de ADE, como medicamentos, efectos adversos e indicaciones [9][10].

Un algoritmo estándar para NER, es el etiquetado secuencial palabra a palabra. Se presentan diferentes complejidades para esta tarea pues es en parte dependiente de la segmentación del texto y de su contexto. Por ejemplo, la entidad "*Luis Vuitton*" puede ser tanto el nombre de una persona, como una organización o un producto, por lo que su entidad real dependerá del contexto [11].

2.3. Preparación del corpus para NER

Como se indicó previamente, para dar solución a la tarea de reconocimiento de entidades, la mayoría de los algoritmos realizan un etiquetado secuencial término a término, para esto es necesario preparar los documentos a etiquetar.

TABLA 2.1. ENTIDADES GENÉRICAS EN NER

Tipo	Tag	Ejemplos	Ejemplo en texto
Persona	PER	Personas, personajes	*"**Turing** es un pionero y genio en la computación"*
Organización	ORG	Compañías, Equipos deportivos	*"**Apple** aumentó sus ganancias"*
Locación	LOC	Regiones, montañas, mares	*"**Isla de Pascua** se encuentra en **Oceanía**"*
Geo-política	GEO	Países, estados, provincias	*"Se observó un aumento de inmigrantes en **Concepción**"*
Construcciones	FAC	Puentes, edificios, aeropuertos	*"El **Puente Juan Pablo II** sufrió daños estructurales"*
Vehículos	VEH	Aviones, trenes, automóviles	*"Su protagonista manejaba un **DeLorean**"*

Fuente. Adaptación de ejemplo en inglés, de "Speech and Language Processing" 81[18]

A. Definiciones previas

Algunas definiciones previas [20]:

(i) **Documento:** archivo de texto.

(ii) **Corpus:** conjunto o colección de documentos.

(iii) **Palabras únicas:** se refiere a los tipos de términos presentes en un corpus.

(iv) **Token:** se refiere a cada término presente en un corpus. Por ejemplo, el fragmento "*el perro y el gato*" cuenta con cuatro palabras únicas "{*el*, *perro*, *y*, *gato*}". Es importante indicar que se refiere a términos y no solo palabras, es decir, es posible considerar como token a los signos de puntuación, o a expresiones conocidas (ejemplo: "*Los Ángeles*"), como un solo término.

(v) **Lemma:** se refiere a la raíz léxica de una palabra. Por ejemplo, las palabras "*gato*", "*gatos*" y "*gatuno*" comparten la raíz léxica gato.

(vi) **Stem:** se refiere a la raíz morfológica de una palabra. Por ejemplo, para el ejemplo anterior, "*gato*", "*gatos*" y "*gatuno*" comparten la raíz morfológica "*gat*". Es importante notar que, en este caso, la raíz "*gat*" no corresponde a una palabra válida de la lengua española.

B. Normalización del texto

Asumiendo la existencia de un corpus a utilizar para extracción de información, la **normalización del texto**, implica una serie de pasos:

(i) **Segmentación a nivel de Documento:** Un documento puede contar con varias secciones de texto, pero no necesariamente todas son de interés. Por ejemplo, en el análisis de una publicación científica, este documento puede contener segmentos de *abstract, introducción, metodología, resultados, conclusiones* y otros metadatos como *título, autores y referencias,* los cuales pueden estar indicados por *headers* de *XML* u extraídos mediante *expresiones regulares* (se abordará posteriormente). Puede que el análisis se enfoque sólo en el *título* y el *abstract* de cada artículo, lo que implicaría que éstos serían los segmentos a analizar por documento. En particular para reportes clínicos e indicaciones médicas, estos pueden contener *headers* y *segmentos* propios de la especialidad o de la institución. Es importante tener esto en consideración, pues, por ejemplo, no es lo mismo identificar un medicamento en un segmento de historial

de una anotación médica, que identificarlo como el medicamento recetado para el tratamiento actual [19].

(ii) **Segmentación a nivel de oración (o sentencia):** Se considera a la oración como la unidad estándar para el análisis de documentos de textos [19]. Una segmentación superficial utilizaría simplemente la puntuación para realizar este procesamiento, pero es evidente de ver que existen casos especiales, como, por ejemplo, "*Mr. Smith*" o "G*eorge R. R. Martin*". Existen algoritmos implementados de segmentación como NLTK (Natural Language Toolkit) [21], disponible para Python, que consideran estos casos especiales.

(iii) **Tokenización:** Posterior a la segmentación, se segmenta a nivel de token, que se considera la unidad a etiquetar posteriormente en la tarea NER. La puntuación y caracteres especiales pueden ser considerados o eliminados, de acuerdo a las características específicas del dominio (por ejemplo, en el ámbito químico se buscará mantener la expresión "*H+*" como un único token para ser identificado como un ión y no confundido con "*H*") [19]. Generalmente, en este procesamiento se puede normalizar el texto a minúsculas. En este punto pueden utilizarse diccionarios para realizar extensión de contracciones ("*don't*" a {"*do*", "*not*"}) y/o realizar corrección de ortografía [19]. Es posible utilizar algoritmos de *minimum edit distance*, como el *algoritmo de Levenshtein* para identificar de forma automática errores de ortografías, alineando la palabra no reconocida con un diccionario de palabras válidas y reemplazándola por la más cercana [20].

(iv) **Part of Speech Tagging (POS):** También conocido como etiquetado gramatical, asigna a cada token una descripción de su rol dentro de la oración. Se basa en las ocho principales entidades del lenguaje en inglés: *sustantivo, verbo, pronombre, preposición, adverbio, conjunción, participio y artículo*. Las implementaciones de POS disponibles actualmente, incluyen además las múltiples variaciones de cada tipo de entidad gramatical (por ejemplo, *VB* para verbo base "*eat*", *VBD* para verbo en pasado "*ate*" y *VBG* para el verbo en gerundio "*eating*"), además de tag especial para puntuaciones [22]. NLTK cuenta con una implementación de POS Tagging, pre-entrenado [21]. Sin embargo, es importante indicar que el etiquetado gramatical es muy dependiente del dominio, en particular para el ámbito médico, en la tarea POS para

documentos clínicos, un etiquetador puede tener desempeños distintos para documentos de diferentes centros médicos [19].

Otros procesamientos que pueden ser aplicados pueden ser: *Parsing* (o análisis sintáctico), que busca separar frases o conjuntos de palabras dentro de una oración que se encuentran separados por conjunciones; *Desambiguación,* para reemplazar diferentes sinónimos por un solo concepto que les represente o entregar el significado correspondiente de acuerdo al contexto; *Temporalidad*, identificación del orden de eventos de segmentos de texto (en el caso de detección de efectos adversos por drogas, es importante determinar qué medicamento se administró previo al evento y cuál en respuesta); y, *Detección de Negaciones*, el análisis por ventanas de texto puede provocar falsos positivos, por ejemplo, "*sin fiebre o escalofríos*" es sinónimo de "sin fiebre" y "*sin escalofríos*", pero en el fragmento original "*escalofríos*" no cuenta con la negación directa "*sin*", por lo cual puede provocar un falso positivo del evento adverso o síntoma que en realidad no presenta [19].

C. Representación del Entidades para NER

El etiquetado de entidades relevantes en el texto es un tipo de tarea de entrenamiento supervisado, pues se trabaja sobre un corpus con textos anotados por expertos donde las entidades se asumen como los verdaderos positivos. Como se mencionó previamente, las entidades pueden tener extensión de una o más palabras. Para el entrenamiento y ajustes del algoritmo de aprendizaje, las entidades son representadas mediante ***BIO tagging***:

- ***B*** de *beggining* (del inglés, "*comienzo*") al primer token de una entidad.
- ***I*** de *inside* (del inglés, "*dentro*") a los tokens de la misma entidad, desde el segundo en adelante.
- ***O*** de *outside* (del inglés, "*fuera*"), a los tokens no etiquetados como entidad.

Mediante esta forma, es posible distinguir dos entidades del mismo tipo adyacentes, pero puede penalizar en un aumento significativo de las etiquetas por entrenar. Este número de etiquetas para un numero inicial de "n" etiquetas, queda definido por (1). Si esta situación es escasa o inexistente, es posible representar mediante una versión simplificada, llamada ***IO tagging***, en la cual no se distingue si una palabra es la primera o no de una entidad. De esta forma, se logra una disminución en el total de etiquetas a entrenar, número definido por (2). A modo explicativo, se presenta un ejemplo para el texto "…*a unit of AMR Corp. immediately…*" en TABLA 2.2 [18], donde sólo se considera una entidad anotada "***AMR Corp***", con la etiqueta ***[ORG]***.

$$\#\ etiquetas\ BIO = \mathbf{2n + 1} \qquad (1)$$

$$\#\ etiquetas\ IO = \mathbf{n + 1} \qquad (2)$$

TABLA 2.2. EJEMPLO DE ETIQUETADO BIO E IO

Palabra	Etiqueta BIO	Etiqueta IO
of	O	O
AMR	B-ORG	ORG
Corp	I-ORG	ORG
,	O	O
inmediately	O	O

Fuente. Adaptación de ejemplo en inglés, de "Speech and Language Processing" [18]

D. Representación del texto para NER

Una vez que se cuenta con un texto tokenizado y asociado a su respectiva etiqueta, es necesario definir las características con las que se representará a la palabra. En este caso, POS tagging fue presentado previamente, a este se suman [18]:

(i) ***Chunk Tagger*:** Realiza un etiquetado por fragmento, entrega información adicional al POS tagging, identificando segmentos de pronombre o segmentos de verbo, y entregando una codificación BIO (TABLA 2.3).

(ii) ***Shape*:** Entrega unformación de la forma de la palabra, reemplazando mayúsculas por el carácter '*X*' y minúsculas por '*x*'. Su versión reducida es llamada *Short Shape* (TABLA 2.3).

(iii) **Prefix y Sufix:** Incluye información de posibles prefijos y sufijos de la palabra. Para hacerlo de forma automática, se determina un umbral, por ejemplo, menor o igual a cuatro caracteres y se incluyen como característica (TABLA 2.3).

Luego, estas características son utilizadas como input para el algoritmo de clasificación, (ver Fig. 2.1).

TABLA 2.3. EJEMPLO DE CARACTERÍSTICAS O FEATURES

Palabra	POS	Chunk	Shape	Short Shape	Prefix largo ≤4	Sufix largo ≥4	BIO tag
of	IN	B-PP	xx	x	'o'	'f'	O
AMR	NNP	B-NP	XXX	X	'A', 'AM'	'MR', 'R'	B-ORG
Corp	NNP	I-NP	Xxxx	Xx	'C', 'Co', 'Cor'	'orp', 'rp', 'p'	I-ORG
,	,	O	,	,	-	-	O
inmediately	RB	B-ADVP	xxxxxxxxxxx	x	'i', 'in', 'inm', 'inme'	'tely', 'ely', 'ly', 'y'	O

Fuente. Adaptación de ejemplo en inglés, de "Speech and Language Processing" [18]

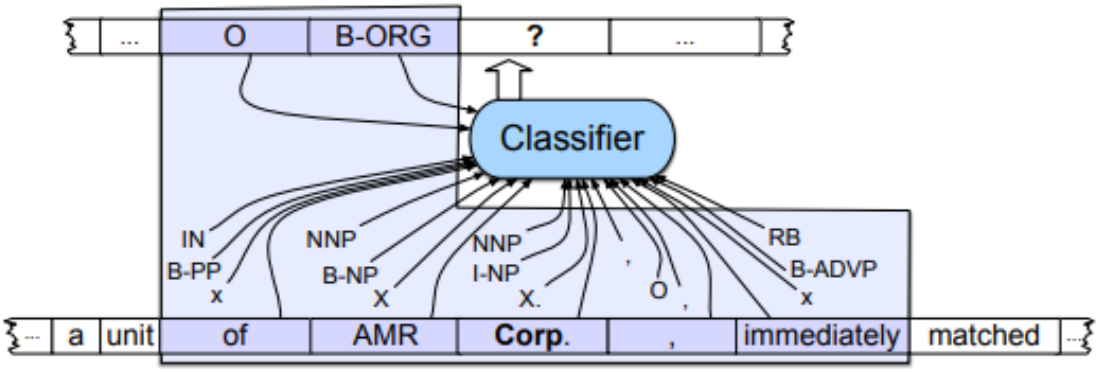


Fig. 2.1 Ejemplo conceptual de algoritmo de clasificación.
Fuente. "Speech and Language Processing" [18]

Las palabras, caracteres y las etiquetas son representados en forma numérica mediante un *one-hot-vector* o *one-hot*-encoding, definido en forma vectorial de acuerdo a (3).

$$\vec{x} = [0\ 0\ 0 \ldots 0\ 0\ 1\ 0\ 0 \ldots 0\ 0\ 0\ ] \quad (3)\ donde \begin{cases} x_i = 1, \ \ si\ i = j\ para\ C_j \\ \\ x_i = 0, para\ i \neq j\ para\ Cj \end{cases}$$

Donde Cj corresponde a la etiqueta o entidad representada.

2.4. Evaluación de tarea NER

Para la evaluación de esta tarea, al tratarse de un problema de clasificación supervisado, se utilizan las métricas de precisión, recall y F-measure, las cuales son definidas para clasificación binaria, en la cual existe una sola clase de interés y los elementos pueden pertenecer o no a esta. A partir de lo anterior se definen los siguientes conceptos [19]:

(i) **TP:** True Positives o "*Verdaderos Positivos*" para una *Clase* o *Entidad*, corresponden al número de tokens clasificados **correctamente** como esta Entidad.

(ii) **FP:** False Positives o "*Falsos Positivos*" para una *Clase* o *Entidad*, corresponden al número de tokens clasificados **incorrectamente** como esta *Entidad* cuando correspondían a otra o a ninguna.

(iii) **FN:** False Negatives o "*Falsos Negativos*" para una *Clase* o *Entidad*, corresponden a al número de tokens clasificados **incorrectamente** como *"O" ("output")* u *otra Entidad*.

A. Precision, Recall y F-measure para evaluación de Clasificación Binaria

Se define como **Precisión (Pr)** a la medida de predicción positiva, definido por (4), entrega la proporción de términos etiquetados correctamente, con respecto al total de términos etiquetados como *esa Entidad*.

Se define por **Recall (Re)** a la medida de sensibilidad del clasificador. Definido por (5), entrega la proporción de términos etiquetados correctamente, con respecto al total de términos que correspondían verdaderamente a esa Entidad.

$$Pr = \frac{TP}{TP + FP} \qquad (4)$$

$$Re = \frac{TP}{TP + FN} \qquad (5)$$

Finalmente, se define como **F-measure** a una métrica ponderada entre Precisión y Recall. Esta es definida por (6), dependiente de un parámetro β. Cuando $\beta = 1$, se habla de ***F1-score*** o simplemente ***F1***, definido por (7) [22].

$$F_\beta = (1 + \beta^2)\frac{Precision \cdot Recall}{\beta^2 \cdot Precision + Recall} \quad (6)$$

$$F_1 = 2\frac{Precision \cdot Recall}{Precision + Recall} \quad (7)$$

Para NER, es importante indicar que estas mediciones se realizan para entidades que pueden contener más de un token. Por lo que entrenar por token, pero evaluar a nivel de entidades puede causar desajustes en las métricas [19].

B. Micro y Macro average para evaluación de Clasificación Multiclase

NER, es una tarea de clasificación multiclase (en general, considera múltiples *Entidades*) por lo que se obtendrán métricas de Pr, Re y F1 para cada una de las clases por separado. Para evaluar el desempeño del sistema completo, se describen tres tipos de promedio; *macro, micro y weightened-average* [23] [24]. A continuación se describirán estos promedios, a partir del ejemplo de la Fig. 2.2, que corresponde a una clasificación multiclase de correos electrónicos en tres clases: urgente, normal y spam. Se presenta la matriz de confusión de la clase real (*true*) versus la entregada por el sistema (*system*) por cada clase y en conjunto del sistema completo (*pooled*).

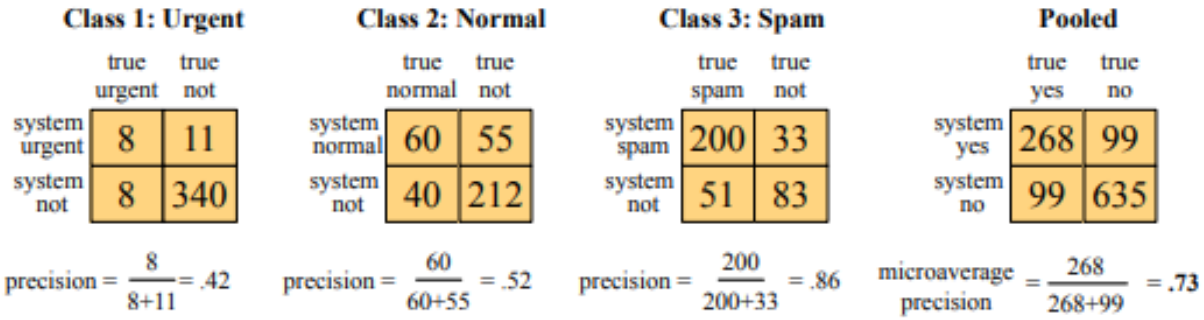


Fig. 2.2 Ejemplo sistema de clasificación de 3 clases.
Tablas de contingencia por clase y del sistema completo.
Fuente. "Speech and Language Processing" [23]

(i) **Macro-average:** Corresponde a un promedio de la métrica sobre el número de clases. Por ejemplo, para el caso de clasificación de 3 clases en Fig. 2.2, la ***Pr-macro*** estaría definida por la ecuación (8). Considerando C_k como clase k, de un total de K clases:

$$\underset{\text{macro}}{\text{Pr}} = \frac{\sum_k \text{Pr}_\text{k}}{\text{K}} = \frac{0.42 + 0.52 + 0.86}{3} = 0.60 \quad (8)$$

(ii) **Micro-average:** Corresponde a un cálculo sobre una tabla de contingencia en la que se suman los *TP, TN, FP* para todas las clases sobre la cual se determina la métrica. Para el ejemplo anterior, ***Pr-micro*** es definido por (9).

$$\underset{\text{micro}}{\text{Pr}} = \frac{\sum_k \text{TP}_\text{k}}{\sum_k (TP_k + FP_k)} = \frac{8 + 60 + 200}{(8 + 11) + (60 + 55) + (200 + 33)} = \frac{268}{367} = 0.73 \quad (9)$$

(iii) **Weightened-average o promedio ponderado:** Corresponde a un promedio ponderado de acuerdo al porcentaje que representa cada clase. Para el ejemplo utilizado, ***Pr-ponderado*** estaría definido por (10).

$$\underset{\text{ponderado}}{\text{Pr}} = \sum_k w_k \cdot \text{Pr}_\text{k} = \left(\frac{16}{367}\right) \cdot 0.42 + \left(\frac{100}{367}\right) \cdot 0.52 + \left(\frac{251}{367}\right) \cdot 0.86 = 0.74 \quad (10)$$

(i) **Inverse weightened-average o promedio ponderado inverso:** Debido a nuestro interés por dar más importancia a las clases menos frecuentes, se define un promedio inverso, donde se pondera cada clase de forma inversa a su frecuencia. El ***Pr-inverso*** estaría definido por (10).

$$\underset{\substack{\text{ponderado} \\ \text{inverso}}}{\text{Pr}} = \sum_k \frac{(w_k)^{-1}}{\sum (w_k)^{-1}} \cdot \text{Pr}_\text{k} = \left(\frac{23}{28}\right) 0.42 + \left(\frac{3.7}{28}\right) \cdot 0.52 + \left(\frac{1.5}{28}\right) \cdot 0.86 = 0.46 \quad (10)$$

Es importante considerar que, en el caso de existir un balance entre las clases, las métricas convergerán a un mismo valor. En el caso de desbalance de clases, las métricas micro y ponderada tenderán a mostrar el comportamiento sobre la entidad que tenga más instancias en el corpus.

C. Validación Cruzada.

Para validar el sistema entrenado, tradicionalmente se utilizan tres sets o grupos de documentos, con el objetivo final de evitar un sobreajuste del modelo a los datos de entrenamiento y que este responda correctamente a nuevos datos [23]:

(i) **Training Set (Set de Entrenamiento):** Utilizado para entrenar al algoritmo o modelo.

(ii) **Development test set (devset):** Es utilizado como set de prueba durante entrenamiento. Permite determinar el comportamiento a datos no vistos durante el entrenamiento. Es utilizado para evaluar diferentes modelos y ajustar hiperparámetros del sistema.

(iii) **Test set (Set de Prueba):** Este set no es presentado al sistema durante el entrenamiento, ni es utilizado para determinar hiperparámetros. Es utilizado para reportar el rendimiento del sistema.

En forma tradicional, estos tres sets pueden ser fijos (ejemplo, 80%, 10% y 10% para (i), (ii) y (iii), respectivamente), en general, buscando utilizar la mayor parte de los datos disponibles como entrenamiento. Esta aproximación puede provocar que el *devset* y *Test Set*, sean poco representativos [23]. Otro método de evaluación, es el uso de ***k-fold cross-validation***, en éste, un porcentaje de los datos es utilizado como *Test Set,* mientras el resto es separada en *k folds* o sub-conjuntos generados aleatoriamente. En la Fig. 2.3, se presenta un ejemplo ilustrativo, con $k=10$ (*10-fold cross validation).* En ésta, se realizan *10* iteraciones de entrenamiento en las cuales se utiliza un subconjunto distinto como *devset* y los restantes como *Training Set*, mientras tanto un conjunto de *Test Set*, dejado aparte para ser utilizado en la evaluación del sistema [23]. Existen implementaciones para realizar validación cruzada y evaluación de métricas multiclase, como lo es el módulo *Scikit-Learn*, disponible para *Python* [24].

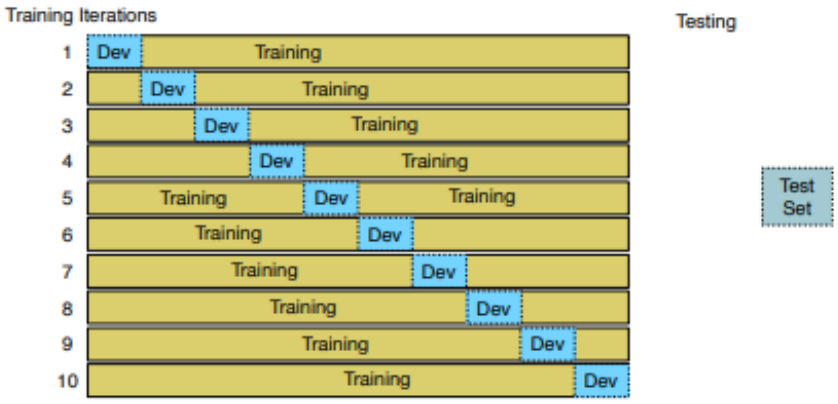


Fig. 2.3 Ejemplo de 10-fold cross-validation.

Fuente. "Speech and Language Processing" [23]

2.5. Métodos de Extracción para NER

2.5.1 Sistemas basados en reglas

Desde la academia, el uso de modelos secuenciales de origen estadístico se observa como norma para la tarea NER en extracción de información. Sin embargo, la industria asociada a la extracción de entidades se mueve por un camino tradicional utilizando principalmente *modelos basados-en-reglas* (*rule-based information extraction*) [25]. Como su nombre lo indica, este tipo de sistemas utiliza una lista de reglas de alta precisión generados por expertos, que son identificados mediante expresiones regulares o simples "*matches*" de cadenas de texto (*strings*). Este tipo de métodos puede ser asociado posteriormente a un etiquetador probabilístico utilizando los tags anteriores como información adicional [18]. Si bien el uso de reglas generales o heurísticas asociadas al dominio (*Knowledge Engineering*) puede ser agregado al procesamiento del corpus, en general, desde la academia el uso de estos sistemas es percibido como un método poco desafiante, lo que no impide que sea el método líder para la industria [25].

2.5.2 NER como problema de clasificación:

Debido a que la tarea NER implica asociar una entidad correcta a cada palabra dentro de una lista de entidades predefinidas, ha permitido que se trate como un problema de clasificación multiclase [26]. Desde este enfoque se le han dado soluciones típicas para este tipo de problemas, como el uso de *Support Vector Machine* (SVM *o máquina de soporte vectorial)*. Este tipo de algoritmo realiza la búsqueda de un hiperplano que optimice la separación de las clases, maximizando vectores de soporte que corresponde a la distancia de los puntos en los bordes cercanos al hiperplano [27][28].

2.5.3 NER como etiquetado secuencial probabilístico:

Desde un punto de vista probabilístico, existen diferentes aproximaciones.

A. Hidden Markov Model (HMM).

Hidden Markov Model (HMM) o "Modelo Oculto de Markov" es un modelo probabilístico que busca maximizar la probabilidad conjunta entre la secuencia observada y la secuencia de etiquetas. Para poder tratar este problema, HMM realiza dos simplificaciones [29]:

(i) **Propiedad de Markov:** Cada etiqueta (t_i) sólo depende del estado anterior (t_{i-1}).

(ii) Cada palabra observada (x_i) sólo depende de la etiqueta (t_i).

$$P(t, x) = P(x|t)P(t) \qquad (11)$$

Donde:

$$x = (x_1, x_2, \dots, x_n), \textit{una secuencia de palabras de largo } n.$$
$$t = (t_1, t_2, \dots, t_n) \textit{ una secuencia de u de entidades } t$$

Ambas simplificaciones permiten tratar a (11) de acuerdo a (12). Esto se puede observar en forma conceptual con el modelo gráfico probabilístico de HMM de la Fig. 2.4.a. En este modelo las flechas del modelo gráfico indican las dependencias de cada término. Por lo que, en este caso, cada palabra "x_i" depende sólo de su etiqueta "t_i", y a su vez, cada etiqueta depende sólo de la etiqueta anterior. Estas probabilidades son calculadas a partir del propio corpus [29].

$$P(\boldsymbol{t}, \boldsymbol{x}) = \prod_{i=1}^{n} P(x_i|t_i)P(t_i|t_{i-1}) \qquad (12)$$

Para responder mejor a nuevas observaciones, el término $P(x_i|t_i)$ se representa mediante la ecuación (13). Esta descomposición se realiza asumiendo independencia de las "j" características utilizadas para representar la palabra "x_i" (representadas por el término f_{ij}) y mediante Naïve Bayes [26].

$$P(x_i|t_i) = \prod_{j} P(f_{ij}|t_i) \qquad (13)$$

B. Maximum Entropy Markov Models (MEMMs)

HMM posee algunas desventajas al modelar el problema mediante un modelo generativo de probabilidad conjunta, cuando en realidad es un problema condicional ante una secuencia de palabras entregada. Además, al utilizar la simplificación aplicada en (12) y (13), HMM asume independencia

entre las palabras (y asume una independencia entre las características o *features* que las describen). Ante estas desventajas, se propuso el uso de *Maximum Entropy Markov Models (MEMMs)* que utiliza la probabilidad condicional $P(t|x)$, aproximada por (14) [26].

$$P(\boldsymbol{t}|\boldsymbol{x}) = \prod_i P(t_i|t_{i-1}, \boldsymbol{x}) \qquad (14)$$

Como se observa en Fig. 2.4.b, este modelo asume que la etiqueta "t_i", solo depende de la etiqueta anterior ("t_{i-1}") y de la palabra "x_i". Sin embargo, este modelo asume dependencia entre el estado actual y el anterior, pero ignora al resto de los estados [26].

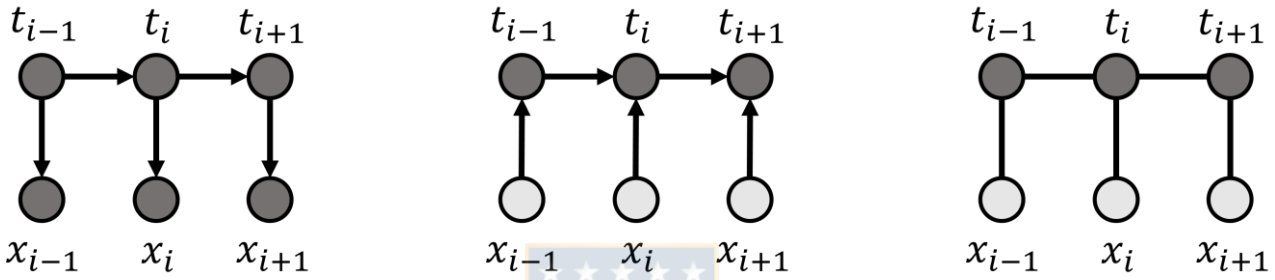


Fig. 2.4 Modelo gráfico de etiquetado secuencial.

(a) HMM. **(b)** MEMM **(c)** CRF
Fuente. Adaptado de Lafferty J. et al (2001) "Conditional Random Fields" [30].

C. Conditional Random Fields (CRF)

Como respuesta a las limitaciones de los modelos anteriores, se presenta *Conditional Random Fields* (CRF) o "*Campo Aleatorio Condicional"* en 2001 [30]. CRF es un modelo discriminativo entrenado para maximizar la probabilidad condicional $P(t|x)$. Puede ser visto como un modelo gráfico no direccionado, de acuerdo a Fig. 2.4.c. e intuitivamente se puede entender como un modelo que utiliza tanto las características que describen a t_i a partir de $\boldsymbol{x}$, como las que describen a las etiquetas t_{i-1} y t_{i+1} a partir de $\boldsymbol{x}$, para determinar t_i.

En este modelo, la distribución condicional es representada por la multiplicación de las funciones de características ("*f"* y *"g"*), de acuerdo a (15). Esta aproximación disminuye las posibles combinaciones y permite representar más información [29].

$$P_\theta(\boldsymbol{t}|\boldsymbol{x}) = \frac{1}{Z_0}\exp\left(\sum_{i=1}^{N}\sum_{k=1}^{M}\lambda_k f_k(t_{i-1}, t_i, \boldsymbol{x}) + \sum_{i=1}^{N}\sum_{k=1}^{M}\mu_k g_k(t_i, \boldsymbol{x})\right) \qquad (15)$$

Donde:

Z_0, factor de normalización de todas las secuencias de etiquetas.

$f_k(t_{i-1}, t_i, \boldsymbol{x})$, función de características asociadas a la etiqueta "t_i", la etiqueta anterior "t_{i-1}" y a la secuencia de palabras.

$g_k(t_i, \boldsymbol{x})$, función de características asociadas a la etiqueta "t_i" y a la secuencia de palabras.

λ_k, μ_k, pesos de aprendizaje para cada función.

N, número de elementos de la secuencia $\boldsymbol{x}$.

M, número de características.

Finalmente, se busca determinar los parámetros $\theta = (\lambda_1, \lambda_2, \ldots, \lambda_M; \mu_1, \mu_2, \ldots, \mu_M)$ a partir de los datos, que maximicen la función objetivo "O" de *log-likelihood* (verosimilitud logarítmica) definida por (16). Considerando un set de datos de entrenamiento $D = \{(x_1, t_1), \ldots, (x_N, t_N)\}$ [29][30].

$$O(\theta) = \sum_{i=1}^{N} \log\left(P_\theta(\boldsymbol{t}^{(i)} | \boldsymbol{x}^{(i)}\right) \qquad (16)$$

2.5.4 Deep Learning

Existen múltiples definiciones para el concepto de *Deep Learning*, pero la mayoría acuerda en que se trata de un campo de *Machine Learning* basado en algoritmos capaces de modelar problemas en abstracciones de múltiples capas y generalmente asociado al uso de redes neuronales multicapa o redes neuronales profundas [35]. Algunas de las soluciones actuales a la tarea NER se han realizado con el uso de *Deep Learning*, específicamente con el uso de redes neuronales recurrentes [9][10]. A continuación, se presentarán algunos conceptos básicos asociados a la definición, diseño y entrenamiento de este tipo de modelos.

A. Perceptrón

El perceptrón es la unidad básica de una red neuronal. Esta estructura fue definida en la década de los 50 [35] y posee cierta inspiración en la biología, en el funcionamiento de la neurona, la cual es capaz de recibir múltiples entradas (*inputs*), cuya sumatoria es recibida por una función de activación que definirá la salida de la unidad. Una red neuronal artificial está formada por múltiples perceptrones interconectados [36]. En la Fig. 2.5, se definen diferentes elementos para un perceptrón.

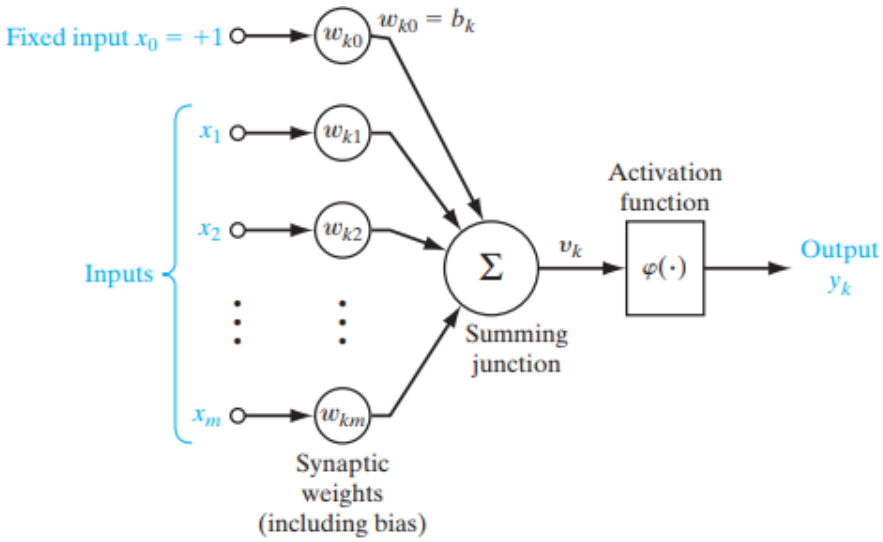


Fig. 2.5 Modelo de perceptrón.
Fuente. Haykin S. (2009) "Neural Network and Learning Machines" [36].

En primer lugar, sus entradas o *inputs* x_i. En segundo lugar, se define un peso (*weigth*) w_{ki}, que corresponde a un factor de multiplicación asociado a cada entrada *i*, este valor es ajustado mediante la presentación de datos de entrenamiento. Se considera un *bias* b_k, que correspondería al peso w_{k0} para una entrada fija definida como $x_0 = 1$ [36] [37].

Luego, la sumatoria de la multiplicación de las entradas y sus pesos, a la que llamamos v_k, definido por (17), ingresará a la función de activación $\varphi(v_k)$ que determina la salida del perceptrón (ecuación (18)).

$$v_k = \sum_{i=0}^{n} w_i x_i \qquad (17)$$

$$y_k = \varphi\left(\sum_{i=0}^{n} w_i x_i\right) \qquad (18)$$

Considerando al vector de entradas $\boldsymbol{x} = [x_0, x_1, \dots, x_n]$ y el vector de pesos $\boldsymbol{w} = [w_0, w_1, \dots, w_n]$, la salida del perceptrón queda definida en forma vectorial por (19) [36] [37].

$$y_k = \varphi(\boldsymbol{x}^T \boldsymbol{w}) \qquad (19)$$

En redes neuronales clásicas, la función de activación $\varphi(v_k)$ correspondía a un umbral de activación (Fig. 2.6.a) que le asignaba un valor entre *[0,1] o [-1, 1]* [36]. Posteriormente, se utilizaron funciones como la sigmoide (Fig. 2.6.b) o tangente hiperbólica [36]. Redes neuronales modernas, utilizan la función de activación *ReLU* (Rectified Linear Unit) (Fig. 2.6.c) [37].

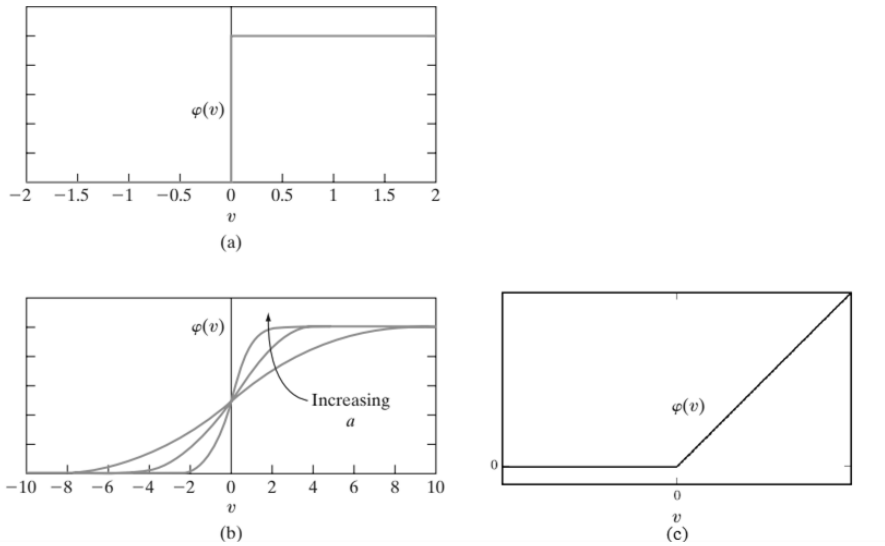


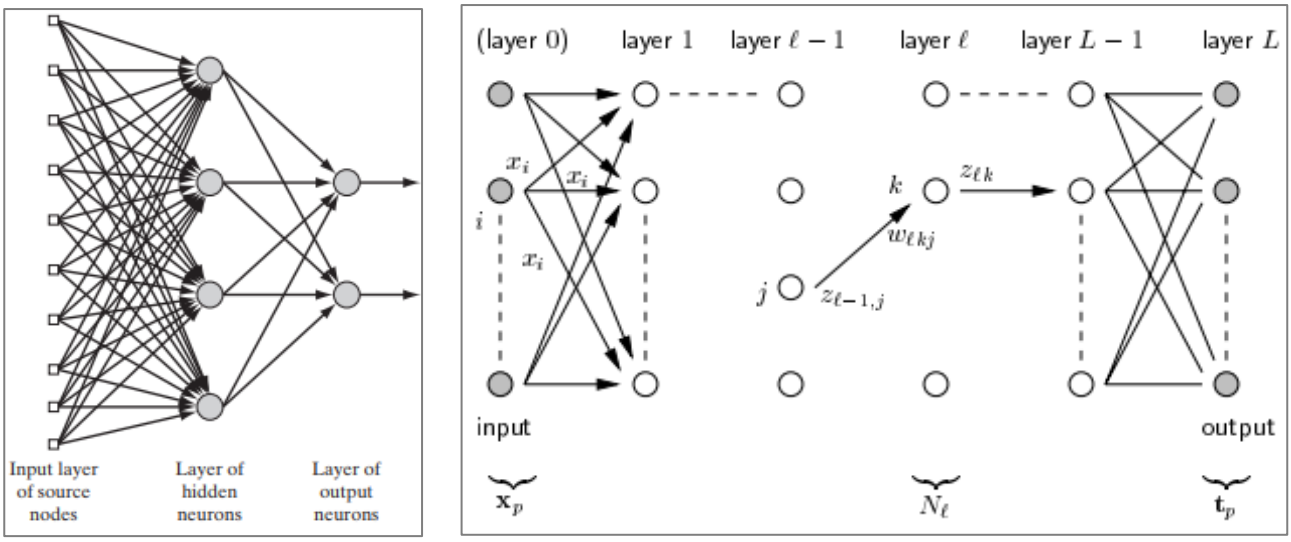
Fig. 2.6 Funciones de activación
(a) Umbral **(b)** Sigmoide **(c)** ReLU
Fuente. (a)(b)Haykin S. (2009) "Neural Network and Learning Machines" [36].
(b) Goodfellow I. et al. (2016) "Deep Learning" [30]

B. Red Neuronal Multicapa

Una red neuronal está formada por capas de uno o más perceptrones y a su vez, de múltiples capas. La cantidad de capas que se asocia a la profundidad de la red [36].

La red neuronal *feedforward* de tres niveles de la *Fig. 2.7.a,* cuenta con una capa de entrada de diez nodos, una intermedia (u oculta) de cuatro neuronas y una capa de salida (*output*) de dos neuronas [36].

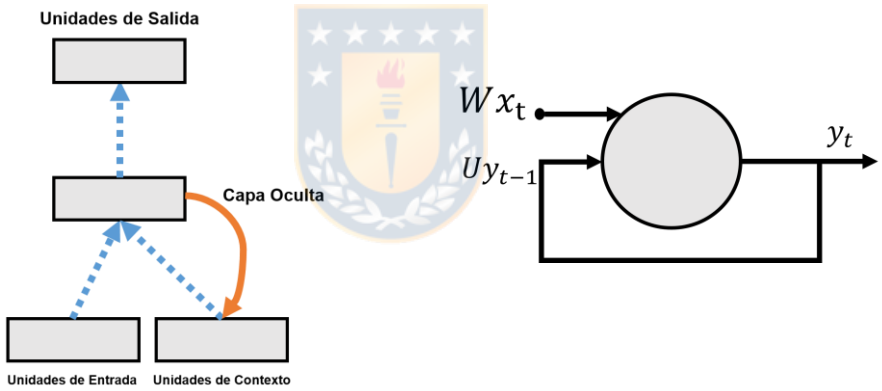
La generalización para *L* capas corresponde a la *Fig. 2.7.b* [28]. En esta red, las entradas corresponden a la primera capa (*layer 0*) de dimensión x_p; las capas ocultas (*layer 1* a *layer L -1*) generalizadas como capa "ℓ" con dimensiones variables y definidas por "$N\ell$"; y, finalmente, la capa de salida u *output* (*layer L*) de dimensión t_p. Las salidas de una neurona "*j*" dentro una capa "ℓ *-1*", (con j < Nℓ *-1*) corresponde a "$z_{j,\ell-1}$". A su vez, el peso asociado a la conexión entre la neurona "$z_{j,\ell-1}$" y la neurona "$z_{k,\ell}$" de la siguiente capa, es expresado como "$w_{\ell,k,j}$". Esta red se considera *feedforward,* pues tiene su entrada desde la capa 1 y salida por la capa "L". Una capa "ℓ" se considerará *fully-connected* si todas sus unidades se conectan con todas las de la capa "$\ell+1$", es decir "$w_{\ell+1,k,j} \neq 0$", para todo "*k*" y "*j*" [28].



(a) (b)

Fig. 2.7 Red Neuronal Multicapa

(a) Feedforward fully-connected **(b)** Red Neuronal Multicapa de "L" capas.
Fuente. (a) Haykin S. (2009) "Neural Network and Learning Machines" [36].
(b) Hristev, R. M. (1998). "The ANN Book" [38]



(a) (b)

Fig. 2.8 Red Neuronal Recurrente

(a) RNN simple definida por Elman (1990) **(b)** Vista a nivel de perceptrón
Fuente. (a) Adaptación propia, Elman, J. L. (1990). "Finding structure in time" [39]. (b) Adaptación propia.

C. Redes Neuronales Recurrentes

Las redes neuronales recurrentes o RNN (*Recurrent Neural Network*), fueron propuestas por primera vez a principio de los años 90's [39]. La RNN simple (Fig. 2.8.a) propuesta por Elman en ese entonces, consideraba una copia directa de la activación de la capa oculta en unidades concatenadas a la capa de entrada y denominadas "*unidades de contexto*". Este tipo de redes están formadas por un nuevo tipo de perceptrón (Fig. 2.8.b) que acepta dos tipos de entrada; la actual y la salida previa de la unidad. De esta forma, la salida del perceptrón en un momento "$\boldsymbol{t}$" es denominada "$\boldsymbol{y_t}$" y queda definida por la ecuación (20), donde "$\boldsymbol{x_t}$" son las entradas a la neurona en ese momento (asociadas a

sus pesos "$\boldsymbol{W}$"), e "$\boldsymbol{y_{t-1}}$" corresponde a la salida previa (asociada a sus propio factor de peso "$\boldsymbol{U}$"). La idea de esta implementación es que la red sea capaz de encontrar dependencias en secuencias de tiempo o en secuencias de texto (en el caso de la tarea NER).

$$\boldsymbol{y_t} = \varphi(W\boldsymbol{x_t} + U\boldsymbol{y_{t-1}}) \qquad (20)$$

D. Long Short-Term Memory (LSTM)

Las redes LSTM (*Long Short-Term Memory*) fueron definidas a mediados de los 90's [40], introduciendo el concepto de celdas de memoria (*memory cells*) y compuertas de activación (*gated units*). En palabras simples, una celda de memoria puede almacenar información importante para "recordar" en el futuro, pero debe "olvidar" con el tiempo la información que ya no sea relevante. Su memoria se debe "proteger" de entradas muy diferentes a lo visto previamente, y a la vez, su salida no debe afectar "agresivamente" a la propia celda, ni a las siguientes. Este tipo de redes nace en respuesta a la "pérdida de memoria" de las RNN para dependencias de largo plazo [40][41].

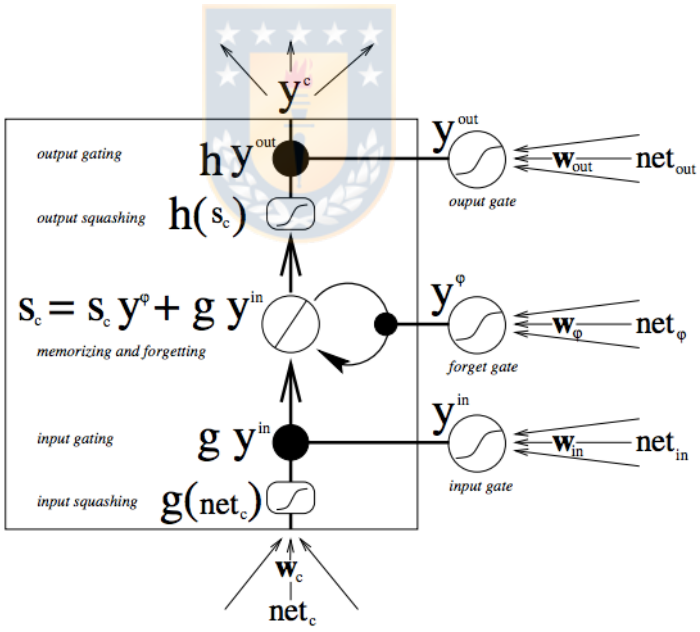


Fig. 2.9. Celda de memoria de LSTM.
Fuente: DL4J, (2017) "A Beginner's Guide to Recurrent Networks and LSTMs" [41]

Una celda de memoria LSTM (Fig. 2.9.b), está compuesta por [40][41]:

(i) **"$\boldsymbol{y^{in}}$":** (input gate unit) compuerta de entrada (21) que protege la memoria de la celda, posee una función de activación "f_{in_j}", que depende de la salida anterior de la celda "$y^c_{(t-1)}$" y asocia pesos "w_{in_jc}" entrenados para esta tarea

$$y_j^{in}(t) = f_{in_j}\left(net_{in_j}(t)\right) \quad ,donde \quad net_{in_j}(t) = \sum_u w_{in_jc} y^c_{(t-1)} \quad (21)$$

(ii) **"y^{out}":** (output gate unit) compuerta de salida (22), que protege a otras celdas de ruido o información irrelevante de la celda actual.

$$y_j^{out}(t) = f_{out_j}\left(net_{out_j}(t)\right) \quad ,donde \quad net_{out_j}(t) = \sum_u w_{out_jc} y^c_{(t-1)} \quad (22)$$

(iii) **"y^{φ}":** (forget gate unit) compuerta de memoria (23), entrega un factor de olvido sobre el estado anterior de la celda.

$$y_j^{\varphi}(t) = f_{out_j}\left(net_{\varphi_j}(t)\right) \quad ,donde \quad net_{\varphi_j}(t) = \sum_u w_{\varphi_jc} y^c_{(t-1)} \quad (23)$$

(iv) **"y^{c}":** (cell output) salida de la celda, definida por (24), donde $s_c(t)$ corresponde al estado actual de la celda en el tiempo *t*. Este estado se actualiza en cada iteración considerando la compuerta de memoria y la de salida.

$$y^{c_j}(t) = y_j^{out}(t) \cdot h\left(s_{c_j}(t)\right) \quad (24)$$

Considerando:

$$s_{c_j}(0) = 0$$

$$s_{c_j}(t) = y_j^{\varphi}(t) \cdot s_{c_j}(t-1) + y_j^{in}(t) \cdot g\left(net_{c_j}(t)\right) \qquad para\ t > 0 \quad (24)$$

$$donde, \qquad net_{c_j}(t) = \sum_u w_{c_jc} y^c_{(t-1)}$$

E. Entrenamiento y aprendizaje de redes neuronales

Las redes neuronales son entrenadas a través de *batches*, que corresponden a sets de entrenamiento sobre los cuales se determina la actualización de los pesos de la red. La actualización para un peso w_{ij} en una iteración "$n+1$" es definida mediante (25), lo que implica que el nuevo valor w_{ij}, será descrito por (26):

$$\Delta w_{ij}^{(l)}(n+1) = -\eta \delta_j^{(l)}(n) y_i^{(l-1)}(n) \qquad (25)$$

$$w_{ij}^{(l)}(n+1) = w_{ij}^{(l)}(n) - \eta \delta_j^{(l)}(n) y_i^{(l-1)}(n) \qquad (26)$$

Donde, el término δ_j corresponde a la retropropagación del error (*backpropagation algorithm*) a partir del método de gradiente descendente, definido por (27); η a la constante de aprendizaje e $y_i^{(l-1)}$, que corresponde a la activación asociada al peso actualizado [36][37].

$$\delta_j^{(l)}(n) = \begin{cases} \dfrac{\partial J^{(L)}(n)}{\partial \varphi_j\left(v_j^{(L)}(n)\right)} \cdot \dfrac{\partial \varphi_j\left(v_j^{(L)}(n)\right)}{\partial v_j^{(L)}(n)} & ,para\ neurona\ j\ en\ capa\ de\ salida\ L \\ \varphi'_j\left(v_j^{(l)}(n)\right) \sum_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n) & ,para\ neurona\ j\ en\ capa\ oculta\ l \end{cases} \quad (27)$$

Para determinar $\delta_j^{(l)}$, se define la recurrencia definida en (27). Para la neurona "j" de la capa de salida "L", el error dependerá de una función de costo "$J_j^{(L)}$" y su diferencial con respecto a salida de esa neurona; y, a su vez, de la diferencial de la salida, con respecto a $v_j^{(L)}(n)$.

Para la neurona "*j*" de la capa "*l*", el error dependerá de $\varphi'_j\left(v_j^{(l)}(n)\right)$, que corresponde a la diferencial de la salida de la neurona "*j*" con respecto a su argumento; y de la sumatoria de los errores de la capa siguiente a los que se encuentra conectada la neurona "j" a través de los pesos $w_{kj}^{(l+1)}$.

El aprendizaje de la red se realiza en épocas (*epoch*), que corresponde a la propagación *feed-forward* de un *batch* y la retroprogación de su error; y en *batches*, que corresponde al entrenamiento por subconjuntos del set de entrenamiento. Se espera ajustar los pesos (*w*) a aquellos que minimicen una función de costo *J* definida en (28), a partir de una función de pérdida *Loss*, que corresponde a una medida del error entre la salida de la red ($y^{(i)}$) para una entrada ($x^{(i)}$) y la salida esperada, para un *batch* de "*m*" elementos.

$$J(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} Loss(x^{(i)}, y^{(i)}, \boldsymbol{w}) \quad ,para\ un\ batch\ de\ m\ ejemplos \quad (28)$$

Para evitar el sobre-entrenamiento, existen métodos de regularización como *early stopping* (se detiene el entrenamiento cuando el error sobre el conjunto de validación comienza a aumentar, con respecto al error sobre el conjunto de entrenamiento), o agregan un factor de regularización a la función de costo [36][37].

Otro método de regularización es *dropout,* que podría traducirse como "dejar fuera". Este método de regularización "apaga" de forma aleatoria un porcentaje de neuronas de una capa, de acuerdo a un porcentaje definido. De esta forma, la red puede hacerse más robusta al tener que

aprender, en el fondo, mediante a diferentes arquitecturas (o arquitecturas variables durante el entrenamiento), evitando la co-adaptación y co-dependencia entre neuronas de una capa y otra [42].

Redes neuronales modernas utilizan *batch normalization*, (que podría traducirse como normalización de batches) y como su nombre lo indica, busca que el entrenamiento sea independiente de los cambios en la varianza que existe entre mini-batches. Esta normalización permite el uso de constantes de aprendizaje mayores y le hace menos dependiente de la inicialización [43].

2.6. Representación Vectorial del Texto

En este capítulo se revisaron diferentes formas de representación de texto como *POS-tagging, chunking* y *shap*e. Para el entrenamiento de redes neuronales, se suele utilizar una representación vectorial del texto. Esta forma de representación se basa en encontrar un espacio latente de dimensionalidad suficiente para representar las palabras como un punto dentro de este espacio, y que entregue información semántica de la misma. En otras palabras, que conceptos con significados similares se aglomeren en este espacio.

Tradicionalmente, *Latent Semantic Analysis* (LSA), también denominado *Latent Semantic Indexing* (LSI), obtiene una representación vectorial a partir de una descomposición lineal, específicamente, mediante el método de *singular value decomposition* (SVD), ejemplificado en la Fig. 2.10 y definido por (29). La representación se obtiene a partir de la matriz de *frecuencia de términos por documento* (A_k) considerando "m" documentos y un vocabulario de palabras únicas de dimensión "n"; las matrices ortogonales *U* (*matriz de términos*) y *V* (*matriz de documentos*), y la matriz diagonal de *valores propios* $\sum$. Luego, es posible representar tanto a términos como documentos en un espacio reducido (y aproximado) de *k* dimensiones, donde $k<r$, sobre el cuál se pueden aplicar *querys* (consultas) o usar como *features* (características) [44].

$$A_k = U\Sigma V^T \qquad (29)$$

Es importante notar que este método está basado en una *matriz de frecuencia de términos por documento*, por lo que estará muy ajustada al documento sobre el cual trabajar y significa un tratamiento de una matriz de alta dimensionalidad y dispersa, pues, generalmente, no se presentan todas las palabras del vocabulario en todos los documentos, por lo que su frecuencia de aparición en estos casos es cero. Hoy en día, existen métodos de representación vectorial llamados "*Word-Embeddings*" que utilizan otros acercamientos modernos como *Deep Learning,* y disponen de versiones pre-entrenadas para representación de términos de uso general.

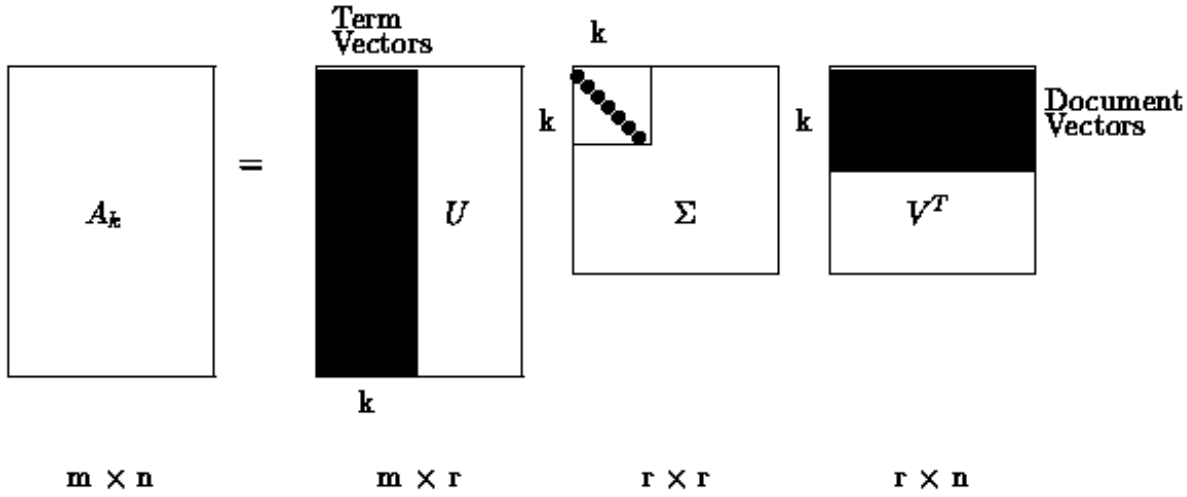
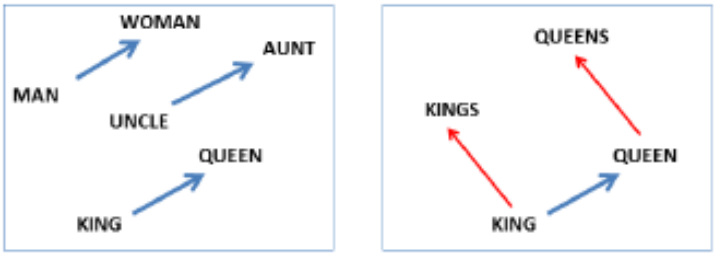


Fig. 2.10 Latent Semantic Analysis.
Fuente. Berry M. (1995) "Computational Methods for Intelligent Information Access" [44].

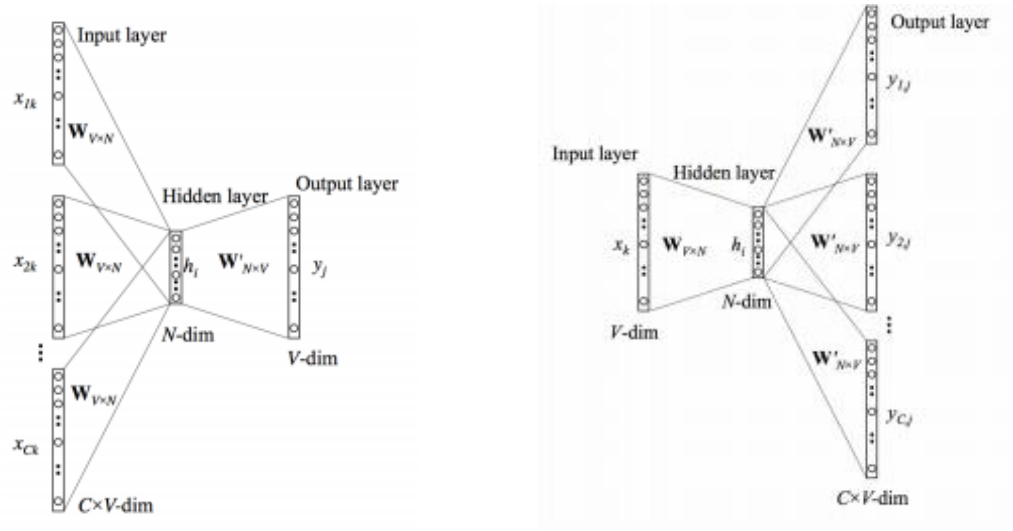
2.6.1 Word-Embedding

Word-Embedding representa en forma vectorial la semántica de cada palabra en un espacio latente de vectores con valores reales [45]. Existen diferentes modelos, tales como CBOW (*continuous bag-of-words model*) [46], *skip-gram* [47] (entrenados con modelos de *Deep Learning*, y presentados en 2013) y GloVe *(Global Vector)* [48] (entrenado a partir de un análisis de coocurrencias de palabras, por ventanas). Algunos análisis realizados a estos modelos, permiten demostrar que las representaciones vectoriales de las palabras cumplen con regularidades lingüísticas (Fig. 2.11). Estas regularidades se observan como relaciones geométricas entre los puntos en este hiper-espacio, como distancia entre el masculino y femenino de un término (representado por el vector azul en la Fig. 2.11.a), o como la distancia entre el singular y plural de un término (representado por el vector rojo en la Fig. 2.11). Los modelos descritos anteriormente, utilizan como entrenamientos corpus de uso general como Wikipedia [46][47][48].



(a) (b)

Fig. 2.11 Relación geométrica entre términos en *Word-Embedding*
(a) relación de género. **(b)** relación singular/plural
Fuente. Mikolov T., et al (2013) "Linguistic Regularities in Continuous SpaceWord Representations" [49]



(a) (b)

Fig. 2.12 Modelos de Word-Embedding por Redes Neuronales
(a) CBOW. **(b)** Skip-Gram
Fuente. Meyer D., (2016) "How exactly does word2vec work?" [50]

A. CBOW y Skip-Gram

Ambos modelos fueron presentados en 2013, por *Mikolov et al.*[46], para su uso como representación vectorial de texto en google utilizando *Deep Learning*. Para su entrenamiento se utilizó un vocabulario de 10.000 palabras únicas representadas por un *one-hot-encoding* (vector de ceros con solo una posición en 1 que representa la posición de la palabra codificada dentro del vocabulario), como capa de entrada a una capa oculta de 300 unidades sin función de activación ($\varphi(x) = x$), y, posteriormente, a una capa de salida de 10.000 unidades con función de activación *softmax* [46].

En forma general, *CBOW* es entrenado para predecir una palabra a partir de la bolsa de palabras vecinas dentro de una ventana de contexto. Esto se observa en *Fig. 2.12.a*, donde, en la iteración "*k*", el modelo recibe "*C*" vectores de entrada "*x*" codificados por un *one-hot-encoding* con dimensión "*V-dim*", que corresponde al tamaño del vocabulario. Posteriormente esta entrada se transmite a la capa oculta de dimensiones "*N-dim*" y se entrenarán los pesos "W_{VxN}" y "W'_{NXV}", para predecir a la palabra deseada en la capa de salida, por lo que se tratará de una capa de "*V-dim*" con activación *softmax* [46].

Por otro lado, *Skip-Gram* es entrenado para predecir las palabras del contexto a partir de una palabra dada "x", lo cual se observa en la *Fig. 2.12.b*, donde se realiza un modelo análogo al de *CBOW*, pero en el cual la capa de salida busca predecir a las palabras "*y*" del contexto de "*x*" [47].

Para el entrenamiento se utilizan los pares de palabras dentro de la ventana. Ambas tareas de entrenamiento son en realidad "*falsas*", pues al finalizar, solo se extraerán los 300 valores de la capa oculta, por palabra, que consistirán en los vectores de 300 dimensiones para representar los términos.

Algunas características de estos modelos, es que pueden ser entrenados por *bi-grams* (pares de palabras) o *frases*, además de no necesitar *stemming* (acortar una palabra a su raíz semántica), pues palabras similares tenderán a tener contextos similares y posteriormente vectores cercanos [47].

2.7. Clustering

Un algoritmo de clustering es un método de aprendizaje no supervisado que permite agrupar un grupo de elementos en forma automática, basándose en métricas de similitud y distancia, pudiendo ser de tipo jerárquico o no jerárquico. Los algoritmos de tipo jerárquico asignan pertenencia a un cluster, como por ejemplo el algoritmo jerárquico aglomerativo asigna un clúster a cada elemento y luego comienza a agruparlos entre sí de acuerdo a alguna medida de distancia. De esta forma, un mismo elemento puede pertenecer a varias agrupaciones y, generalmente, es representado a través de un diagrama de árbol o dendrograma. En un algoritmo no jerárquico, el espacio es dividido en grupos o particiones que no comparten elementos entre sí, por lo que cada elemento solo puede pertenecer a una agrupación. [51][52]

A. K-means

K-Means (o K-Medias) es un algoritmo tradicional de clústering no jerárquico, basado en particionar los datos en K grupos. Cada grupo es descrito por un valor medio o centroide $\boldsymbol{\mu_k}$ (Fig. 2.13.a), que es reasignado en cada paso del algoritmo hasta converger al valor que logre minimizar la distancia entre los elementos de cada grupo a su centroide (Fig. 2.13.i). [51][52]

Considerando un dataset de N elementos $X = \{x_1, x_2, \dots, x_N\}$, y el indicador $r_{nk} \in \{0,1\}$ que será "1" si un elemento x_n pertenece al clúster "k". Utilizando la distancia euclidiana como medida de distancia de los elementos a cada centroide, se podría describir la función de costo a optimizar mediante (30).

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \cdot \|x_n - \mu_k\|^2 \qquad (30)$$

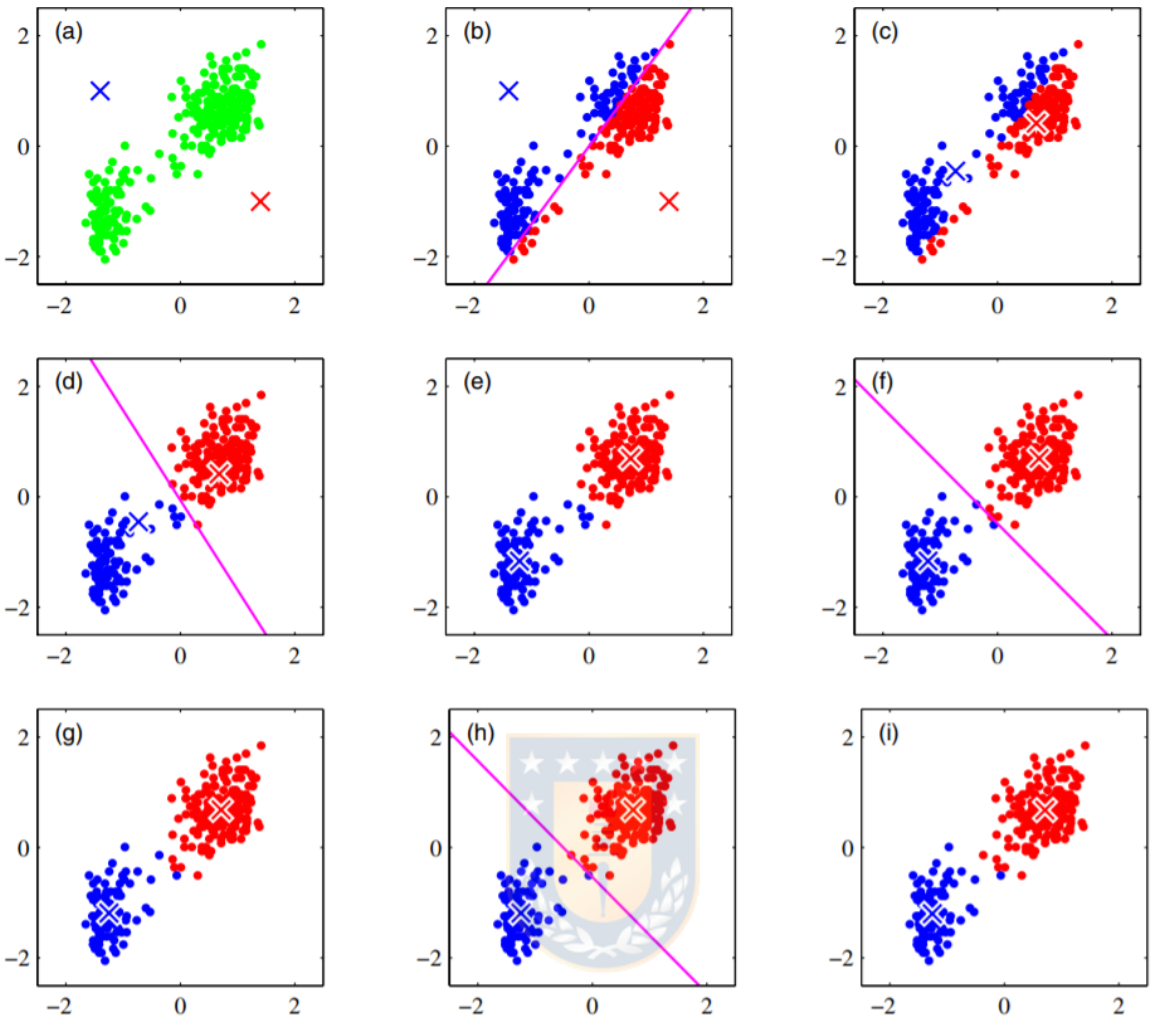


Fig. 2.13 Ejemplo de algoritmo k-means para k = 2 en dataset de 2 dimensiones.
Fuente. Bishop (2006) "Pattern Recognition and Machine Learning". [51][52]

B. Determinar Número de Centroides.

Existen diferentes heurísticas para determinar el número de centroides. En forma intuitiva, se podría pensar en que un buen descriptor de la calidad de un clúster estaría determinado por una mayor similitud intra-cluster y una mayor distancia entre clústers.

El coeficiente de Silhouette [53], permite determinar evaluar el desempeño de un algoritmo de agrupamiento utilizando la distancia promedio entre un elemento x_n y los demás elementos de su clúster "A", definido como $a(x_n)$; y la distancia promedio a todos los elementos del clúster más cercano "B", donde $k \neq B \; y \; x_n \notin B$, definido como $b(x_n)$; finalmente, Silhouette es definido por la ecuación (31).

$$s(x_n) = \frac{b(x_n) - a(x_n)}{\max\{a(x_n), b(x_n)\}} \qquad (31)$$

Se incluye la restricción que para un conjunto de clusters con tamaño 1 (es decir, el caso extremo en que cada elemento sea su propio cluster) se define el valor de $s(x_n) = 0$ [53].

Es posible identificar que el valor de Silhouette variará entre $-1 < s(x_n) < 1$. Tendiendo al valor (-1) cuando x_n tiene una menor distancia con otro clúster que con los elementos del propio (implicando un mal agrupamiento) y tendiendo a $(+1)$ cuando x_n tiene una distancia significativamente mayor con el clúster más cercano que con los elementos del propio (implicando un buen agrupamiento). Siendo 0 cuando $a(x_n) = b(x_n)$.

Finalmente, el Coeficiente de Silhouette de un agrupamiento se definirá como el promedio $\overline{s(x_n)}$ para $x_n \in X$. El número ideal de clústers estará determinado por el número "k" que maximice el valor de $\overline{s(x_n)}$.

Capítulo 3. Estado del Arte

3.1. NER en textos médicos.

Denominado también como BioNER, en el dominio de análisis de textos médicos se asocia de forma general al procesamiento de dos tipos de documentos: artículos científicos del área y archivos clínicos. Ya se presentó previamente que el procesamiento del texto libre no estructurado de anotaciones médicas permite extraer información clínica relevante, sin embargo, posee dificultades propias del dominio. Se podría decir que se trata de un *sublenguaje*, con un vocabulario limitado, con reglas sintácticas limitadas y diferentes a las del lenguaje en general. Por ejemplo, la frase "*No fever or chills*" ("Sin fiebre o escalofríos"), es un fragmento típico de una anotación médica, puede ser reconocido como una oración durante la segmentación, pero al realizar un *POS tagging* no reconocerá verbo, ni sujeto, ni objeto [19]. Pese a la existencia de abundante documentación clínica, el texto libre no suele poseer estándares entre especialidades, ni entre centros clínicos, así como tampoco existe gran cantidad de corpus anotados, lo que dificulta su procesamiento. La tarea NER es el primer paso, fundamental para lograr tareas superiores de IE, como la extracción de eventos, identificando entidades de interés del evento y sus relaciones [54].

3.2. Word-Embedding biomédico

Con el fin de incluir particularidades de textos biomédicos, en 2015 Jiang Z. et al. [45] presentó un *Word-Embedding* para este dominio, considerando *stem*, *chunk* y *entidades biomédicas* (como genes y proteínas). En 2016, Jagannatha A. et al. [10] presentó un *skip-gram biomédico*, entrenado utilizando el corpus de *Wikipedia*, *Artículos científicos de PubMed* y anotaciones médicas de *Pittsburgh*.

3.3. Métodos de Machine Learning en NER biomédico.

Tal como en la tarea NER tradicional, a principios de los 2000 la tendencia en el área fue hacia el uso de HMM (Zhou G. (2002) [31] y Zhao S. (2004) [32]), que cambió hacia el uso de CRF desde mediados de esa década. Algoritmos basados en sistemas de clasificación, se mostraron muy competentes durante este período de tiempo, principalmente en comparación a métodos de Maximización de Entropía como demostró el trabajo de Kazama J. et al. (2002) [28]. Algunas implementaciones de *SVM*, como Lee et al. (2004) [27], presentaron una clasificación en dos etapas:

una para distinguir las palabras *outside* de las pertenecientes a entidades, y luego una de clasificación de entidades dentro de las clases válidas. *Settles B. (2004)* [33] realizó una comparación entre HMM y CRF, demostrando las ventajas y superioridad de CRF para el área de entidades biomédicas, posteriormente Tzong-Han et al. (2006) [26] y *Ponomareva (2007)* [29] utilizaron este método probabilístico, variando principalmente las características de representación de texto, probando la influencia de *POS-tagging* y *steming.* Este método fue aplicado a corpus biológicos con el reconocimiento de entidades como genes y proteínas.

Posteriormente, la masificación del uso de *Deep Learning* para diferentes tareas, se propuso el uso de estos algoritmos para la tarea NER. se han propuesto métodos híbridos con el uso de CRF para uso general y en particular para textos biomédicos. Yao L. (2015)[55], propone el uso de Redes Neuronales Convolucionales o *Convolutional Neural Network* (CNN) para la tarea de BioNER con buenos resultados. Sin embargo, *Huang (2015)* [34] con su trabajo "*Bidirectional LSTM-CRF Models for Sequence Tagging*" demuestra una arquitectura basada en un LSTM bidireccional (utilizando la información de una secuencia en sentidos normal e invertido) en conjunto a CRF, presentando un mejor F1-score frente a otros métodos que utilizaban SVM, y CRF por sí solo, demostrando además buenos resultados pese a variar a las características [34]. En este mismo camino, *Jagannatha, A. N., & Yu, H. (2016)* [9] presentaron una implementación de la tarea NER, para un corpus propio anotado con entidades para la detección de *Eventos de efectos Adversos por Medicamentos* (ADE), se probó el uso de LSTM, LSTM bidireccional y CRF por separado, demostrando que el uso de redes recurrentes para el etiquetado secuencial de textos médicos era posible y superior al desempeño de otros métodos. Posteriormente, el mismo año, presentaron un nuevo trabajo con el uso de una arquitectura de "*Bi-LSTM-CRF*" [10] que superó al método anterior. *Lyu C. (2017)* [56], presentó una comparación del estado del arte de la tarea *BioNER* con Bi-LSTM-CRF en el ámbito de reconocimiento de genes y proteínas, se lograron desempeños comparables al uso de CRF, pero sin la necesidad de recurrir a la generación de reglas y características creadas a mano, sino que utilizando un *Word-Embedding* concatenado a un *embedding* propio a nivel de carácter.

Algunos trabajos, como *Tang B., et al. (2013)* [58], se dedicaron a evaluar la importancia de las características para la representación de textos para *BioNER*. En particular, este trabajo utiliza características de *baseline* un método de CRF, utilizando *POS-Tagging*, sobre las que se probaron tres formas de representación: basada en clustering (se utilizó el *Brown Clustering Algorithm*)[59]; representación distribucional basada en la coocurrencia de términos; y *word-embedding*. Como resultados se demostraron que estas tres lograron un mejor del desempeño que cada una por sí sola,

aumentando el *Recall* y en consecuencia el *F-measure* para los dos corpus analizados. Por su parte, *Habibi M., et al. (2017)* [60], realizó un análisis de las ventajas del uso de *word-embeddings* en BioNER, realizando pruebas sobre 33 corpus (gen/proteína, enfermedades, químicos, entre otros). Se compararon *baselines* disponibles para cada tipo de corpus mediante NER basado en reglas (características tradicionales), método por CRF y método por LSTM-CRF. Este último supero en 5% en *F-measure* a los anteriores. Tanto CRF y LSTM-CRF se consideraron como métodos "agnósticos a las entidades" o "genéricos", utilizando solo un *embedding* concatenando una representación a nivel de palabras (*Word-Embedding*) y a nivel de caracteres (*Character-Embedding*), lo que les permite concluir que el uso de este tipo de técnicas permite desarrollar métodos más robustos y con menores dificultades de desarrollo para adaptarse a nuevos campos.

A. Bidirectional LSTM -CRF

La solución actual desde la aproximación de *Deep Learning*, es el uso de una red neuronal LSTM bidireccional en conjunto a una capa de salida de CRF [9][10][56][60]. En la Fig. 3.1, se muestra un ejemplo de este modelo. La entrada a la red "$\mathbf{X_t}$" corresponde una representación de *word-embedding (*obtenido de una tabla de vectores pre-entrenados*)* y *character-embedding* (entrenado por una capa bidireccional de LSTM considerando la secuencia de caracteres a nivel de palabra). La secuencia de texto es representada como una secuencia de ambos embeddings concatenados, que posteriormente ingresan a una capa LSTM bidireccional y, finalmente, a la capa CRF de salida que generará una secuencia de etiquetas que optimice la probabilidad conjunta.

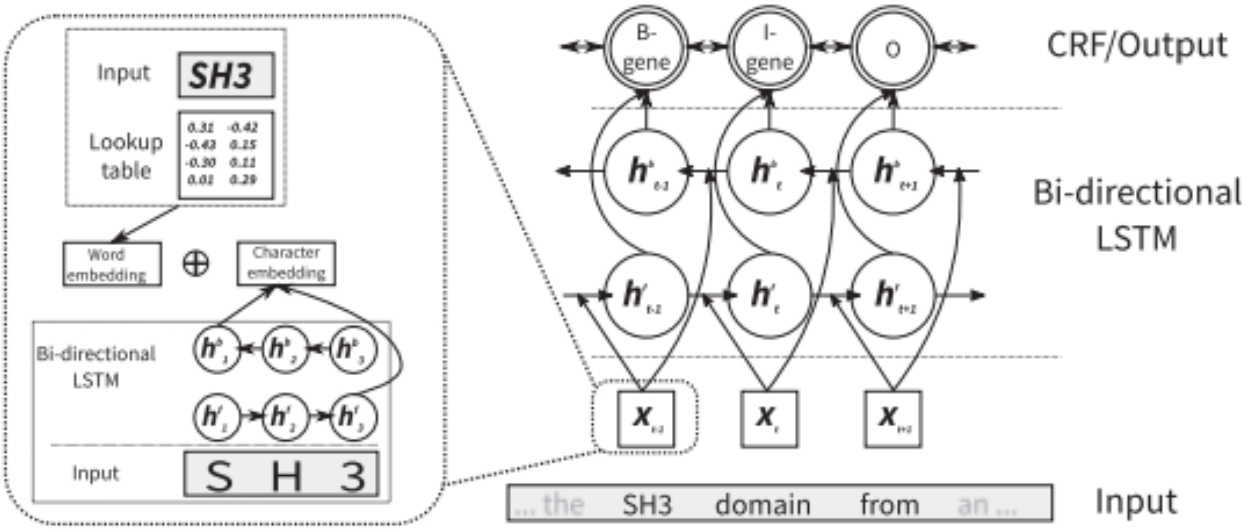


Fig. 3.1 Modelo de Bi-LSTM-CRF
Fuente. Habibi M. (2017) "Deep learning with word embedding improves biomedical NER" [60]

En enero de 2019, los autores del Challenge MADE 1.0, (que prepararon y liberaron el corpus que se utiliza en la presente investigación) presentaron un artículo con un resumen de los principales resultados de los competidores [61]. Específicamente para la tarea NER, todos los competidores usaron el Word-Embedding pre-entrenado por los autores. Algunos de los competidores incluyeron dentro de sus características de entrenamiento terminología clínica de SNOMED-CT (Systematized Nomenclature of Medicine – Clinical Terms), un embedding de sufijos clínicos, *POS (Part-of-Speech)* y *Character-Embeddings.* Como modelo de clasificación, se presentaron modelos de LSTM, BiLSTM y CRF tanto por separado, como combinados. Los mejores resultados fueron obtenidos por el equipo WPI-Wunnava con un modelo BiLSTM-CRF que utilizaba el *Word-Embedding* pre-entrenado en conjunto a un *character-embedding*. Con un Recall de 0.82, Precision de 0.83 y F1-score de 0.82. El método utilizado para el cálculo de estas métricas se realizó por evaluación de "frase-exacta", es decir, solo se consideraban como correctas las entidades identificadas completamente como la entidad correcta sin importar si alguna de las palabras de la frase fue identificada correctamente (a diferencia de esta investigación, en el cual se evaluará la tarea a nivel de palabra).

B. Biomedical NER mediante Transformers

Una tendencia muy interesante en el campo del Procesamiento del Lenguaje Natural y el del Deep Learning, es el uso de Transformers, como lo son BERT (Bidirectional Encoder Representations from Transformers) [62] y GPT-2 [63], los años 2018 y 2019, respectivamente. Estos modelos inicialmente presentados como modelos de lenguaje para traducción y generación de texto demostraron su fortaleza en la habilidad para adaptarse a otras tareas de NLP. Se han presentado algunas aplicaciones enfocadas específicamente en NER (Yan H., 2019) [64] y BioNER (Raza Khan M., 2020)[65] para corpus de origen biomédico para el reconocimiento de genes y proteínas en artículos científicos. Aplicaciones de código abierto como Simple Transformer [66], permiten utilizar modelos pre-entrenados. No se identifican aplicaciones directamente para extracción de efectos adversos (ADE), o utilizando el corpus de MADE. En julio de 2020, este campo se revolucionó con la publicación de GPT-3 de OpenAI [67], un modelo de lenguaje con cerca 175 billones de parámetros (x10 con respecto a GPT-2) que se presentó como un "Few-Shot Learner", pues demostró capacidad de entregar muy buenos resultados para distintas tareas de NLP, a partir de muy pocos ejemplos, generando texto coherente y difícil de distinguir a uno escrito por humanos. Hasta el término de este informe de tesis, este nuevo modelo se encuentra en una beta cerrada, se han dado a conocer diferentes aplicaciones de desarrolladores con acceso a esta beta y los resultados son muy prometedores.

Capítulo 4. Objetivos e Hipótesis

4.1. Problema Identificado y Oportunidad

La información digital está aumentando día a día producto de la digitalización de diferentes servicios [1] e incluyendo al área de la salud [2]. Este fenómeno no es ajeno a la realidad de la salud en Chile, principalmente impulsada por el estado a través del plan de Estrategia Digital [4][5] que han impulsado la implementación de la ficha clínica electrónica y la telemedicina. Con metas de un uso de la ficha electrónica del 100% de los establecimientos al 2020 [6].

Si bien la discusión se ha preocupado principalmente en la implementación e interoperabilidad de la ficha clínica electrónica, en palabras de la Jefe del Departamento de Gestión Sectorial de TIC del Ministerio de Salud; "…*hay que seguir impulsando el uso secundario de información, y abordar la explotación de grandes volúmenes de datos para poder lograr una analítica sanitaria que permita cada vez más soportar la generación de políticas públicas*…" [6].

Chile no se queda ajeno a los errores médicos, una investigación publicada por la Pontificia Universidad Católica en 2008, observó que la tendencia es a un enfoque sistémico, con vigilancia al error médico y a los efectos adversos, además del reconocimiento del error a un paciente consciente a su derecho a ser informado [14]. Existen casos de efectos adversos provocados por administración errónea en hospitales que han provocado, incluso, la muerte [13].

Hoy en día, es posible utilizar la información escrita en lenguaje natural disponible en archivos clínicos electrónicos, para identificar de forma automatizada eventos adversos. La correcta extracción de entidades relevantes [9][10] es el primer paso para una tarea superior, como la extracción de eventos adversos para prevención y farmacovigilancia en centros hospitalarios, cuya identificación no solo es positiva para el desarrollo de políticas públicas o generación de protocolos rápidos y atingentes dentro de los centros clínicos, sino que va en beneficio de la seguridad y atención al paciente.

Se propone introducir mejorar a la tarea de reconocimiento automático de entidades aplicada a *Eventos de Efectos Adversos a Medicamentos (ADE)*, en textos clínicos disponibles en *lenguaje natural* con presencia de un desbalance de clases. Utilizando métodos de aprendizaje no supervisado para la identificación de nuevas entidades en el texto no etiquetado por expertos, utilizando *Deep Learning* para la extracción de características y posterior tarea de clasificación. Este campo se encuentra en pleno desarrollo, aún abierto a posibles innovaciones.

4.2. Objetivos

4.2.1 Objetivo General

Diseñar un sistema para la extracción automática de entidades relacionadas a efectos adversos por medicamentos en textos clínicos, entrenado a partir de un corpus previamente anotado, y basado en técnicas de *Deep Learning.*

4.2.2 Objetivos Específicos

- Estudiar el estado del arte en extracción de entidades para identificar oportunidades de innovación.
- Identificar características que faciliten el proceso de extracción de entidades con algoritmos de *Deep Learning* como, por ejemplo, aquellas obtenidas por *Word-Embedding.*
- Diseñar e implementar un modelo de clasificación secuencial basado en *Deep Learning* que permita obtener un espacio latente de características que faciliten la extracción de entidades médicas relacionadas a efectos adversos.
- Evaluar modelo integrado final en comparación al actual estado del arte.

4.3. Hipótesis

El uso de un espacio latente obtenido del entrenamiento de un modelo de redes neuronales recurrentes sobre secuencias de textos clínicos permitirá descubrir agrupaciones de entidades que ayudarán a mejorar la tarea de extracción automática de entidades médicas, evaluándose en términos del F1-score ponderado sobre el etiquetado de los datos.

4.4. Plan de Trabajo

Para la realización del presente trabajo de investigación de propusieron las siguientes tareas:

I. Realizar revisión bibliográfica de métodos de extracción de información para la tarea NER (Named Entity Recognition) con y sin el uso de Deep Learning
II. Estudiar el estado del arte en referencia a esta aplicación en el área de textos médicos.
III. Caracterizar entidades y atributos del texto del corpus a utilizar.
IV. Diseñar e implementar método de preprocesamiento de textos del corpus.
V. Dividir corpus en set de entrenamiento, validación y prueba
VI. Entrenar modelo basado en redes neuronales para la extracción de entidades.

VII. Aplicar algoritmos de clustering sobre el espacio latente obtenido por el modelo de extracción de entidades.
VIII. Evaluar pertenencia de las clases previamente etiquetadas a los clústeres descubiertos.
IX. Inspección visual de entidades pertenecientes a los distintos clústeres y re-etiquetado.
X. Entrenar nuevamente modelo de extracción de entidades, considerando el re-etiquetado.
XI. Evaluar modelo final a partir de cálculo de métricas de F1-score ponderado sobre las clases de interés y análisis de estos resultados.
XII. Preparación y envío de artículo.
XIII. Redacción de Informe de Tesis.
XIV. Preparación de presentación.

4.5. Recursos Disponibles

Se utilizaron textos de anotaciones médicas en inglés, facilitados por la Universidad de Massachusetts. Estos textos se encuentran de-identificados para proteger la identidad del paciente, y se cuenta con autorización vigente para su uso en tareas de *Machine Learning* y *Text Mining* de acuerdo al IRB (*Institutional Review Board*, en referencia a aprobación por el Comité de Ética) adjunto en el Anexo A.

4.6. Alcances y Limitaciones

La implementación de los modelos de *Deep Learning* se realizará utilizando la librería open-source *Keras* [69], sobre un *backend* de *Tensorflow* [70] en *Python 3.6*. Estas interfaces de programación cuentan con características de alto nivel para la implementación de arquitecturas de redes neuronales, y se encuentran optimizadas para utilizar CUDA, con soporte de cálculo sobre GPU.

4.7. Propuesta de publicación

Durante el desarrollo de esta tesis de Magíster, se propone enviar un artículo científico abarcando los siguientes temas:

- Método de extracción de características para la descripción de secuencias de texto libre en archivos médicos electrónicos.
- Modelo de clasificación de entidades relacionadas a efectos adversos en archivos médicos electrónicos escritos en lenguaje natural.

Capítulo 5. Materiales y Métodos

5.1. Introducción

En el presente capítulo se describe el corpus a utilizar para el entrenamiento y la metodología utilizada para desarrollar, implementar y evaluar la tarea NER.

5.2. Materiales

5.2.1 Descripción de Corpus

La extracción de información para anotaciones médicas se ha convertido en un importante tópico. En este contexto, las universidades de Massachusetts, Lowell, Worcester y Amherst organizaron en 2018 el desafío "*NLP Challenges for Detecting Medication and Adverse Drug Events from Electronic Health Records (MADE1.0)"*[17]. Como parte de este evento, se liberó un dataset de 1089 EHR de-identificados, con el objetivo de implementar un algoritmo automático de detección de entidades médicas relevantes previamente anotadas como ADE, medicamentos, diagnósticos y sus relaciones [17].

```
Patient: [** Name **], [** Name **]       Acct.#:     [** Medical_Record_Number **]       MR
D.O.B:   [** Date **]            Date of     [** Date **]        Location: [** Location **]
                                Visit:
Dictated [** Date **]  8:17 P    Transcribed [** Date **]  9:45
:                               :           P

                                 CLINIC NOTE

DIAGNOSES:
1.  Lymphoplasmacytoid lymphoma involving bone marrow and spleen diagnosed
initially in [** Date **], associated with progressively increasing IgG kappa
paraprotein.
2.  Compression fractures of the spine secondary to lymphoma.  In the past,
it has been associated with significant back pain.  He is status post three
kyphoplasties.  The first in [** Date **] and two in [** Date **].  The first
procedure was complicated by acute hemoglobin decrease for which he was
hospitalized.  Hemorrhagic pericardial effusion was diagnosed and drained.
It was not malignant.  He received 5 units of red cells at that time.
2.  Extended hospitalization in [** Date **].  Then he was admitted for
significant back pain and then developed Salmonella sepsis with necrotizing
fasciitis of right gastrocnemius, which required debridement.  He had a
residual ulcer on the medial malleolus of the right ankle, which is now
fully healed.  He has required several hospitalizations for recurrent
```

Fig. 5.1 Ejemplo de formato del corpus
Texto de documento en ASCII
Fuente. MADE 1.0 (2018) Bio-NLP Lab University of Massachusetts [17]

```
<?xml version="1.0" encoding="ISO-8859-1"?>
- <collection>
    <source/>
    <date/>
    <key/>
  - <document>
      <id>1_9</id>
    - <passage>
        <offset>0</offset>
      - <annotation id="591">
          <infon key="type">SSLIF</infon>
          <location offset="3589" length="6"/>
          <text>fevers</text>
        </annotation>
```

Fig. 5.2 Ejemplo de formato de anotaciones
Fuente. MADE 1.0 (2018) Bio-NLP Lab University of Massachusetts [17]

La Fig. 5.1, muestra el ejemplo de un extracto de una anotación médica perteneciente al corpus. El documento se encuentra de-identificado, por lo cual alguna información sensible como el nombre, número del reporte, fecha y locación, se encuentras reemplazadas por expresiones regulares de la forma "[** …**]". A su vez, se observa que el documento posee una cabecera de información estructurada, que fue eliminada, pues no entrega mayor información para la tarea de reconocimiento de entidades. El resto de la anotación corresponde a texto libre en lenguaje natural, el cual corresponde a informes de diferente extensión en un formato principalmente formal, con el uso de abreviaciones médicas y mínimos errores ortográficos.

La Fig. 5.2, corresponde a un extracto del documento de anotaciones. Este archivo es único para cada documento de texto médico y cuenta con las entidades anotadas y sus relaciones. Este se encuentra en un formato XML denominado BIOC, el cual posee una librería en Python para la extracción de los objetos "entidades" y "relaciones". El archivo es estructurado primero a nivel de colección (<*collection*>), con información adicional de fuente, fecha de creación y key (<*source/*>, <*date/*>, <*key*/>) y posteriormente a nivel de documento (<*document*>) indicando el id del mismo ("1_9" en este caso) y luego, las diferentes entidades anotaciones con sus relaciones. Cada anotación posee un identificador (<*annotation id = "000"*>), indica el tipo de entidad anotada (<*infon key= "type"*>) e indica la extensión del segmento de texto a través del carácter de inicio mediante un offset (<*location offset*>) y su largo en caracteres (*lenght*). Además, incluye el segmento de texto (<*text*>) de la anotación.

TABLA 5.1. TIPOS DE DOCUMENTOS CLÍNICO.

Tipo de Documento	Cantidad
CHIEF COMPLAINT	23
CLINIC NOTE	**452**
DEAR DOCTOR (E-MAIL)	85
DISCHARGE SUMMARY	56
ENDOCRINOLOGY PROGRESS NOTE	3
HISTORY AND PHYSICAL	17
HISTORY OF PRESENT ILLNESS	38
INPATIENT CONSULT	6
OUTPATIENT CONSULTATION REPORT	48
PATIENT DISCHARGE	47
REHABILITATION SERVICES	2
REPORT OF OPERATION	27
OTHER	72
TOTAL	**876**

Fuente: Elaboración propia a partir de MADE 1.0 (2018) Bio-NLP Lab University of Massachusetts [17]

A nivel de documentos, se realizó un etiquetado manual sobre 876 archivos (conjunto de entrenamiento del Challenge MADE 1.0 [17]) para identificar a qué tipos de documentos clínicos correspondían. Se determinó que la mayor cantidad (51,5%) fueron categorizadas como "Clinic Note", que corresponden a notas médicas completas de un paciente, incluyendo su historial, diagnóstico y tratamiento actual. El resto se divide en diferentes tipos de documentos, como "Chief Complaint" (un resumen de un encuentro médico o del estado de un paciente), "Dear. Doctor" (correos electrónicos de un médico resumiendo el historial clínico de un paciente a un colega), "Discharge summary" – "Patient discharge" (resumen entregados al "dar de alta" a un paciente luego de una hospitalización), "History and physical" (historial previo de un paciente) y "History of present illness" (descripción del diagnóstico y tratamiento actual de un paciente). Hay presencia de otros tipos de reportes posterior a un encuentro médico, intervención, consulta u otro.

Con respecto a las entidades anotadas presentes en este corpus, son sobre 79.000 y correspondientes a tres tipos, que se dividen en:

A. Signos, síntomas y enfermedades (SSD)

- **Eventos SSD:**
  - **(i) ADE:** SSD que es un efecto secundario ante medicación.
  - **(ii) Indication:** SSD que está activamente siendo tratado.
  - **(iii) SSLIF:** Otro SSD, por ejemplo, parte de historial clínico.

- **Atributos SSD:**
  - **(iv) Severity:** Frases que describen la intensidad o severidad de los síntomas. (ejemplo, "*a six on a escale of ten*", "*persistent*")

B. Medicación

- **Eventos de medicación:**
  - **(v) Medication:** Droga, medicamento o procedimiento preescrito.
- **Atributos de medicación:**
  - **(vi) Duration:** ciclos de administración. (ejemplo, "*4 weekly dosis*")
  - **(vii) Dose:** dosis administrada (ejemplo, "*1 capsule*", "*80 mg*")
  - **(viii) Route:** vía de administración (ejemplo, "*transdermal*", "*patch*")
  - **(ix) Frequency:** frecuencia de administración (ejemplo, "*every day*")

El corpus cuenta además con las relaciones entre entidades, siendo un total de 27.326 relaciones, de 7 tipos:

- **(i) "do":** relación de dosis, entre "Medication" y "Dose"
- **(ii) "manner/route":** relación de vía de administración, entre "Medication" y "Route"
- **(iii) "fr":** relación de frecuencia, entre "Medication" y "Frequency"
- **(iv) "du":** relación de duración, entre "Medication" y "Duration"
- **(v) "severity_type":** relación de severidad, entre "Medication" y "Severity"
- **(vi) "adverse":** relación de efecto adverso, entre "Medication" y "ADE"
- **(vii) "reason":** relación de causa de medicación, entre "Indication" y "Medication"

Algunas de las particularidades del corpus son inconsistencias en algunas anotaciones, como algunos conceptos escritos de diferente forma (ejemplo, "p.o", "po" y "p.o."), y doble anotación de algunas entidades, principalmente ADE, Indication y SSLIF (Other SSD) pues su diferencia está más asociada al contexto que al vocabulario.

En Fig. 5.3 se observa un claro desbalance entre las clases, además, estas entidades no tienen un largo fijo, variando entre entidades de un término y frases de múltiples palabras. En la TABLA 5.2, se observa que cerca del 80% del corpus corresponde a palabras no anotadas. Además, se observa la existencia de un desbalance dentro de la extensión de las propias clases, presentando una extensión variable. En particular, se detectó el caso de una entidad de tipo "SSLIF" con 15 palabras.

Labels	Number of Entities	Annotated Words
ADE	1940	3255
Indication	3804	8240
Other SSD	39384	82956
Severity	3908	5069
Drugname	15902	19075
Duration	898	1768
Dosage	5694	11820
Route	2667	2805
Frequency	4806	11400

Fig. 5.3 Estadística de entidades.
Fuente. MADE 1.0 (2018) Bio-NLP Lab University of Massachusetts [17]

TABLA 5.2. DESCRIPCIÓN DE LOS DATOS.

	Largo promedio de entidad ± std	**% de instancias Set de validación**
None	-	**82.72%**
ADE	1.68 ± 1.22	0.35%
Indication	2.22 ± 1.79	1.31%
SSLIF	2.12 ± 1.88	8.62%
Severity	1.27 ± 0.62	0.38%
Drugname	1.21 ± 0.60	2.63%
Duration	2.01 ± 0.74	0.21%
Dose	2.09 ± 0.82	2.06%
Route	1.20 ± 0.47	0.39%
Frequency	2.44 ± 1.70	1.33%

Fuente. Basado en MADE 1.0 (2018) Bio-NLP Lab University of Massachusetts [10] [17]

5.3. Metodología

A. Normalización de corpus

Se realiza normalización del texto, a través de:

(i) **Segmentación a nivel de Documento:** Se extrae el texto de los documentos,

(ii) **Tokenización:** Se eliminan caracteres especiales. Se normaliza el texto a minúsculas, y dígitos son reemplazados por el carácter “d”. Se realiza extensión de contracciones y se tokeniza a nivel de palabra.

B. Preparación del corpus.

Cada documento es preprocesado obteniendo una lista de oraciones. Cada oración es representada en formato de lista de tuplas. Cada tupla asocia un token a su entidad correspondiente, a partir de la lectura del texto en xml.

(1) Segmento de texto:

"it developed an acute hemoglobin decrease for wich…"

(2) Segmento en formato de lista de tuplas:

```
[("it", "O"), ("developed", "O"), ("an", "O"), ("acute", "O"),
("hemoglobin", "B-SSLIF"), ("decrease", "I-SSLIF"), ("for", "O"),
("wich", "O"),…]
```

Fig. 5.4 Ejemplo de preparación de anotaciones del corpus.
Fuente: Corpus MADE 1.0 (2018) Bio-NLP Lab University of Massachusetts [17]

Para la representación de las entidades, se utilizó BIO-Tagging, identificando tokens al principio ("B-"; primer token dentro de una entidad), dentro ("I-"; segundo token en adelate dentro de una entidad) y fuera de las entidades ("O", palabras sin entidad asociada). Se implementó esta preparación a través de un código en Python. En el ejemplo descrito en Fig. 5.4; el segmento de texto es procesado a un formato de lista de tuplas, donde cada palabra es convertida en un token y asociada a su entidad correspondiente. El subsegmento "hemoglobin decrease" es identificado como "SSLIF", entidad correspondiente a "otro tipo de signos, síntomas y enfermedades", siendo "hemoglobin" la primera palabra de la entidad ("B-SSLIF") y "decrease" la segunda y última ("I-SSLIF"). El resto de las palabras no corresponden a ninguna de las entidades previamente anotadas ("O").

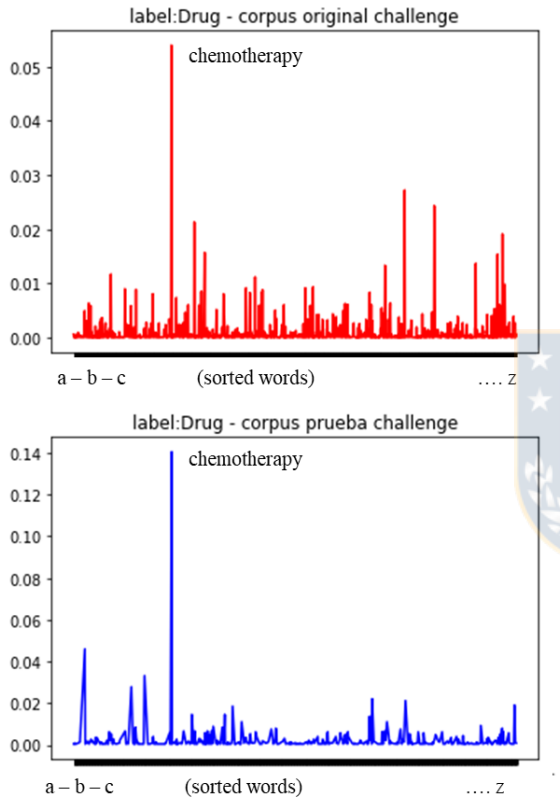
C. Preparación de set de entrenamiento

Para la posterior evaluación del modelo propuesto, se divide el corpus (a nivel de documentos) en conjuntos de entrenamiento, validación y prueba. Del total de 1089 documentos, se utilizó como prueba el conjunto presentado como test en el Challenge MADE 1.0 (2018) [17] correspondiente a 213 documentos. De los 876 documentos de entrenamiento original, se seleccionó una muestra aleatoria de 2/3 (584 documentos) como conjunto entrenamiento, dejando 1/3 restante (292 documentos) como conjunto de validación.

Al evaluar la similitud entre los conjuntos se observa que el desbalance de clases es similar para los conjuntos creados, cumpliéndose que cerca de un 80% corresponde a palabras no etiquetadas y la entidad anotada con mayor presencia en el corpus (~10%) es "SSLIF".

Al evaluar a nivel de bolsa de palabras, se observa una diferencia en la frecuencia de palabras dentro de cada entidad para cada conjunto de documentos. Por ejemplo, analizando el top 15 en frecuencia de palabras para la entidad "Drug" (Fig. 5.5), tanto para en el conjunto original del

challenge (que corresponde a nuestro conjunto de entrenamiento y validación) como para el conjunto de pruebas, la palabra "*chemotherapy*" corresponde a la de mayor frecuencia. Sin embargo, en segundo lugar, para el conjunto de prueba se encuentra "ABVD" [68], un tipo de tratamiento de quimioterapia combinada empleada en tratamiento del cáncer por linfoma de Hodgkin, que no aparece en el top 15 del conjunto original. En la Fig. 5.5 se presenta gráficamente el diccionario de palabras etiquetadas como "Drug", ordenadas en orden alfabético, y es posible visualizar diferencias en la presencia de diferentes términos.



(a)

	0_original	1_freq	2_test	3_freq_test
0	chemotherapy	0.053968	chemotherapy	0.140584
1	prednisone	0.027244	abvd	0.045977
2	rituxan	0.024352	bleomycin	0.033156
3	coumadin	0.021344	bactrim	0.027851
4	vitamin	0.019146	omeprazole	0.022104
5	d	0.015733	prednisone	0.021220
6	velcade	0.015386	zofran	0.019010
7	thalidomide	0.013709	emend	0.018568
8	pamidronate	0.013362	copdac	0.014589
9	antibiotics	0.011742	doxorubicin	0.014589
10	fludarabine	0.011222	oepa	0.013705
11	warfarin	0.009833	ondansetron	0.011052
12	lisinopril	0.009371	etoposide	0.011052
13	lasix	0.009197	paroxetine	0.011052
14	fentanyl	0.009197	tmp	0.009284

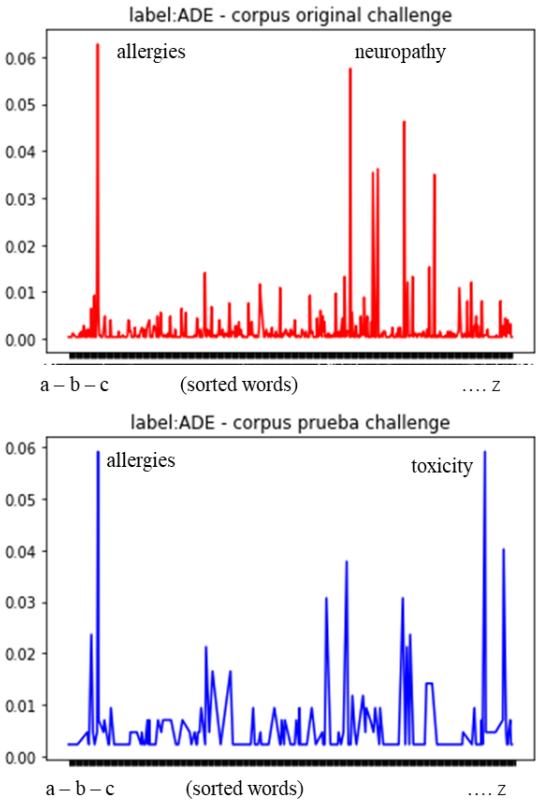
(b)

Fig. 5.5. Frecuencia de palabras para entidad "Drug".

(a) Histograma comparativo de frecuencias normalizadas para dataset de entrenamiento y dataset de prueba. Eje x contiene la lista de palabras del vocabulario ordenados por abecedario, para comparar la variación en la distribución de palabras de ambos dataset.
(b) Top 15 de frecuencia de palabras para entidad Drug, para ambos dataset.
Nota: Drug: (Droga o medicamento). Fuente: Elaboración propia.

	0_original	1_freq	2_test	3_freq_test
0	allergies	0.062701	allergies	0.059102
1	neuropathy	0.057476	toxicity	0.059102
2	rash	0.046222	vomiting	0.040189
3	peripheral	0.036174	nausea	0.037825
4	pancytopenia	0.035370	lung	0.030733
5	skin	0.034968	pulmonary	0.030733
6	pancytopenic	0.016077	reaction	0.023641
7	side	0.015273	adverse	0.023641
8	effects	0.014068	effects	0.021277
9	nausea	0.013264	rash	0.021277
10	renal	0.013264	epistaxis	0.016548
11	reaction	0.012058	fever	0.016548
12	thrombocytopenia	0.012058	side	0.014184
13	his	0.011656	sensitivity	0.014184
14	in	0.010852	neuropathy	0.011820

(a)



(b)

Fig. 5.6. Top 15 en frecuencia de palabras para entidad "ADE".

(a) Histograma comparativo de frecuencias normalizadas para dataset de entrenamiento y dataset de prueba. Eje x contiene la lista de palabras del vocabulario ordenados por abecedario, para comparar la variación en la distribución de palabras de ambos dataset.
(b) Top 15 de frecuencia de palabras para entidad ADE, para ambos dataset.

Nota: ADE: (Evento de efecto adverso por medicamento). Fuente: Elaboración propia

También es posible identificar una diferencia al analizar sobre la entidad con menor presencia en el corpus, pero con mayor importancia clínica: ADE o eventos de efectos adversos a medicamentos. Si bien para ambos conjuntos la palabra más relevante es "*alergies*" (alergias) con una presencia de 6% dentro de las entidades, solo comparten 6 palabras dentro del top-15 de ambos grupos, y en frecuencias diferentes.

Este análisis nos permite visualizar la alta variabilidad del corpus, la riqueza del vocabulario y la complejidad de la tarea de trabajar con texto libre. Más aún, nos permite visualizar la alta dificultad de la tarea NER sobre este corpus, ya que las entidades de mayor interés clínico poseen poca presencia dentro de los documentos y alta variabilidad entre sí, tanto en extensión, como en vocabulario.

D. Representación del texto

Para la representación de las secuencias de texto, se utilizó un *Word-Embedding* pre-entrenado con textos clínicos, el *skip*-gram biomédico de 200 dimensiones entrenado por Jagannatha A. et al. (2016) [10].

E. Entrenamiento de Modelo

El modelo propuesto busca encontrar formas de facilitar la tarea de extracción de entidades en texto de forma automática, para textos que presentan un alto desbalance. Para enfrentar este desbalance, se trabajó realizando clustering a partir del entrenamiento del Estado del Arte (Bi-LSTM-CRF) sobre el conjunto de entrenamiento. Se incluyó una capa densa de veinte dimensiones (de acuerdo al número de clases considerando BIO tagging) previa a la capa de clasificación CRF (Fig. 5.7). Se utiliza **la salida** de la capa añadida para realizar clustering de las **palabras originales**, con el objetivo de asignar una **etiqueta temporal** al 80% de las palabras no etiquetadas del corpus, y considerar un corpus más balanceado para un posterior re-entrenamiento del estado del Arte. Posteriormente, se evalúa el desempeño sobre las **entidades originales**.

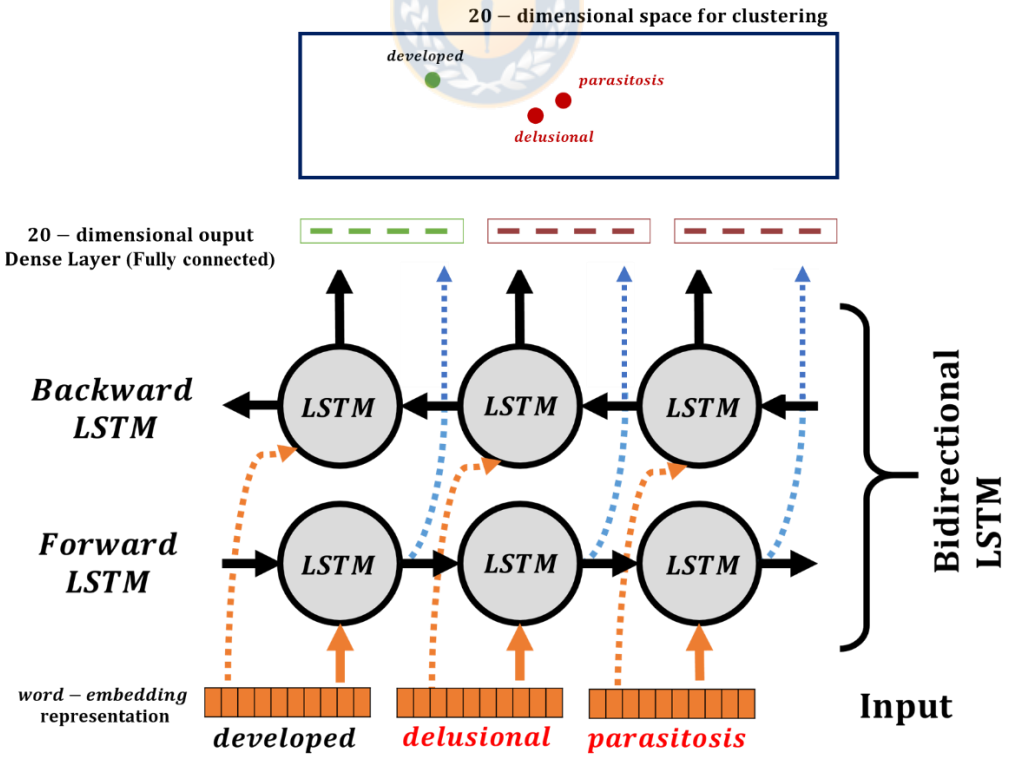


Fig. 5.7. Nuevo modelo propuesto.

Fuente: Adaptación propia

(i) Entrenamiento del estado del arte.

Para la implementación del modelo replicado del estado del arte se utilizó Keras [69], que es una interfaz de programación de alto nivel que trabaja sobre TensorFlow [70] y permite el diseño de arquitecturas de redes neuronales y su entrenamiento tanto en CPU, como acelerado por GPU. Se describe como una interfaz amigable y modular, permite abstraer la arquitectura del modelo donde cada capa, es una línea de código. En Fig. 5.8, se presenta un ejemplo de la implementación de un modelo de red neuronal, entrenamiento y su posterior evaluación. En (Fig. 5.8.a) se realiza la importación de los métodos a utilizar; en (Fig. 5.8.b) se define el modelo de tipo secuencial, con una capa densa de 32 neuronas que considera una capa de entrada de dimensión 784 y una activación de tipo "ReLU". Como se puede observar, la definición del modelo se hace a través de capas, que ya se encuentran predefinidas y las cuales cuentan con parámetros que se pueden modificar; (Fig. 5.8.c) corresponde al entrenamiento del modelo, en este caso, el modelo definido en (Fig. 5.8.b), al cual se le entregan datos (*data*) y sus etiquetas (*labels*), con batches de tamaño 32 y durante 10 épocas; (d) el método *predict*, permite realizar la predicción de etiquetas del modelo para un nuevo set de datos; finalmente, la predicción anterior puede ser evaluada mediante su pérdida (*loss*) y otras métricas (*accuracy*) utilizando (e).

```
from keras.models import Sequential
from keras.layers import Dense, Activation
```
(a)

```
model = Sequential()
model.add(Dense(32, input_dim=784))
model.add(Activation('relu'))
```
(b)

```
model.fit(data, labels, epochs=10, batch_size=32)
```
(c)

```
classes = model.predict(x_test, batch_size=128)
```
(d)

```
loss_and_metrics = model.evaluate(x_test, y_test, batch_size=128)
```
(e)

Fig. 5.8 Ejemplo de uso de Keras.
(a) importación de métodos **(b)** definición de modelo **(c)** entrenamiento **(d)** predicción **(e)** evaluación
Fuente. Keras API Documentation [69].

(ii) Clustering (k-means).

Como algoritmo de clustering se utilizó k-means. Para determinar el número de clústeres se evaluó el coeficiente de silhouette aplicando k-means al grupo de entrenamiento desde 1 a 80 clústers, como se observa en Fig. 5.9.

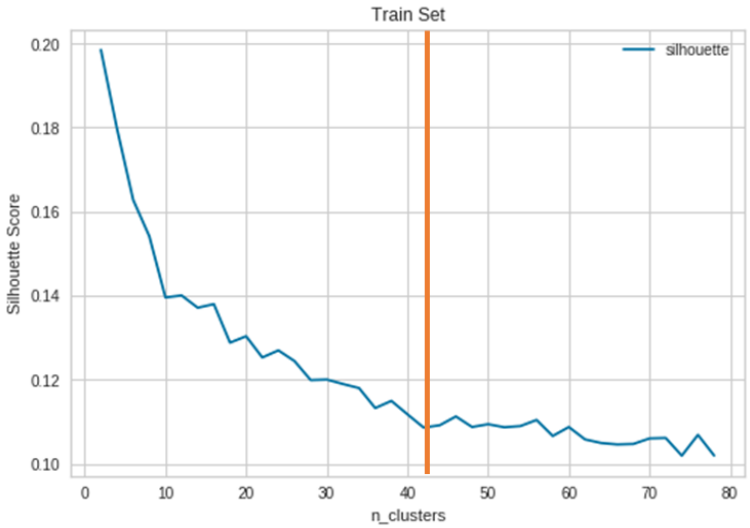


Fig. 5.9 Heurística: "Método del codo".
Nota: Nota: Silhouette Score sobre el total de los datos, para 1 a 80 clústeres. Fuente. Elaboración propia.

Se identificó que el valor más alto de la métrica se obtuvo al considerar un único clúster, desempeño que se ve disminuir al aumentar el número de agrupaciones, hasta llegar a un valle que comienza aproximadamente a partir de los 42 clústeres. Se determina que una sola agrupación no es informativa para la tarea que estamos buscando. Dado que las etiquetas originales son 9, duplicadas a 18 por BIO-Tagging y se espera encontrar un número mayor de agrupaciones a las entidades anotadas. Se decide utilizar 42 clústeres como hiperparámetro para entrenar el algoritmo de agrupamiento.

(iii) Evaluación de pertenencia.

Para evaluar las agrupaciones entrenadas, se visualiza la pertenencia de las clases originales a través de una nube de palabras y mapa de calor. Los resultados se presentarán en el próximo capítulo.

(iv) Re-etiquetado y re-entrenamiento.

Considerando que las entidades de interés representan un 20% del corpus, además de ser etiquetadas por expertos, se consideran estas etiquetas como verdaderas. Para el texto restante, etiquetado como clase "None", se procede a etiquetar utilizando su clase asociada al algoritmo de k-means entrenado. Se considera que el modelo NER posee 60 etiquetas (18 originales + 42 por algoritmo de clustrering). Los resultados se presentarán en el próximo capítulo.

F. Evaluación de Modelo

Como se explicó previamente, se definieron grupos de entrenamiento y prueba. Se utilizó validación cruzada sobre el primero, generando los sub-sets de entrenamiento (*trainset*) y validación (*devset*) para ajustar hiperparámetros del modelo, mediante los métodos de "*train_test_split*", "*cross_validation*" "*classification_report*" y "*confusión_matrix*" de *Scikit-Learn* [24]. Finalmente se entrena un modelo final con el set completo de entrenamiento, y es evaluado sobre el set de prueba.

Se presentarán y analizarán los resultados de *Precision*, *Recall* y *F1-score* a nivel de palabra. Para la evaluación promediada de la tarea NER se utilizará F1-micro, presentando tanto *Pr-micr*o como *Re-micro*. Se considerarán resultados ponderados eliminando a la clase "Output" que permitan interpretar de mejor forma los resultados sobre las entidades de interés, ante el desbalance presente en los datos. Se analizará el impacto del re-etiquetado, sobre las clases de interés con menor presencia en el corpus.

Capítulo 6. Resultados

En el presente capítulo se describen los principales resultados de la actual investigación, de acuerdo con el orden descrito previamente en la metodología.

6.1. Estado del Arte Replicado

A. Entrenamiento del modelo E. del A.

Se replicó el modelo del Estado del Arte, adaptando el modelo Bi-LSTM-CRF descrito por *Jagannatha, A. N., & Yu, H. (2016)* [9] [10] incluyendo una capa fully-connected previa a la capa CRF de salida (Fig. 6.1). La salida de esta penúltima capa es utilizada para el entrenamiento del algoritmo de clustering, de acuerdo con lo descrito previamente en la metodología.

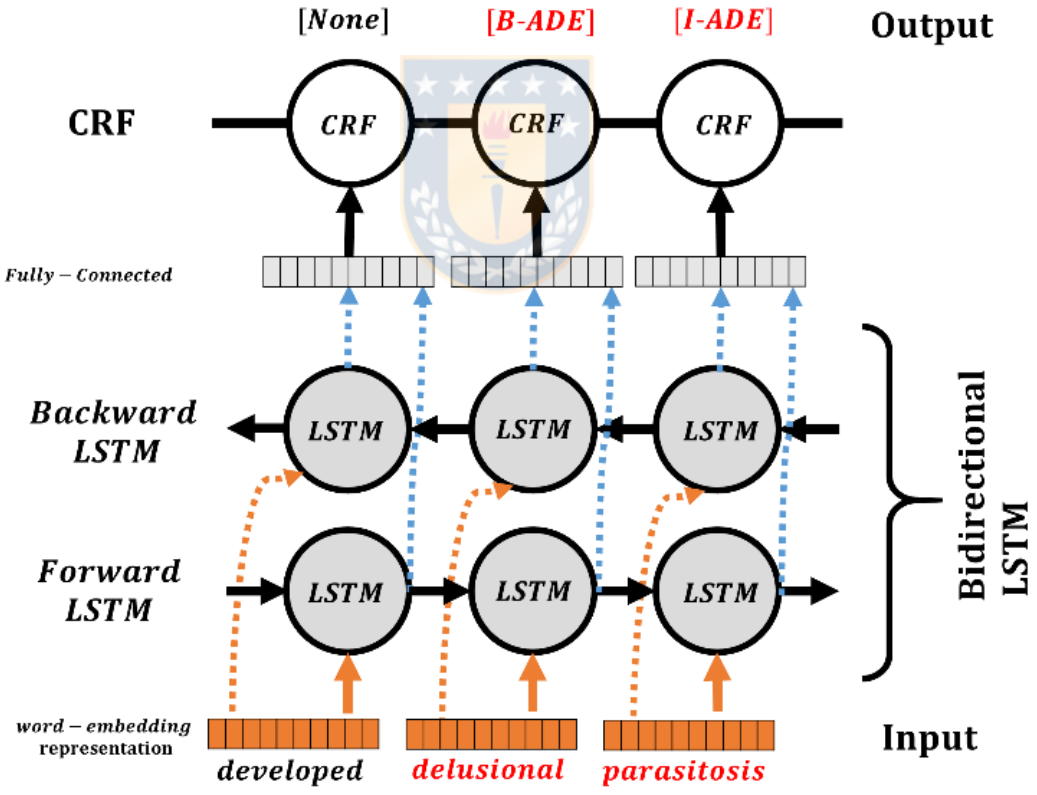


Fig. 6.1. Bidirectional-LSTM-CRF implementado.

Fuente: Adaptación propia

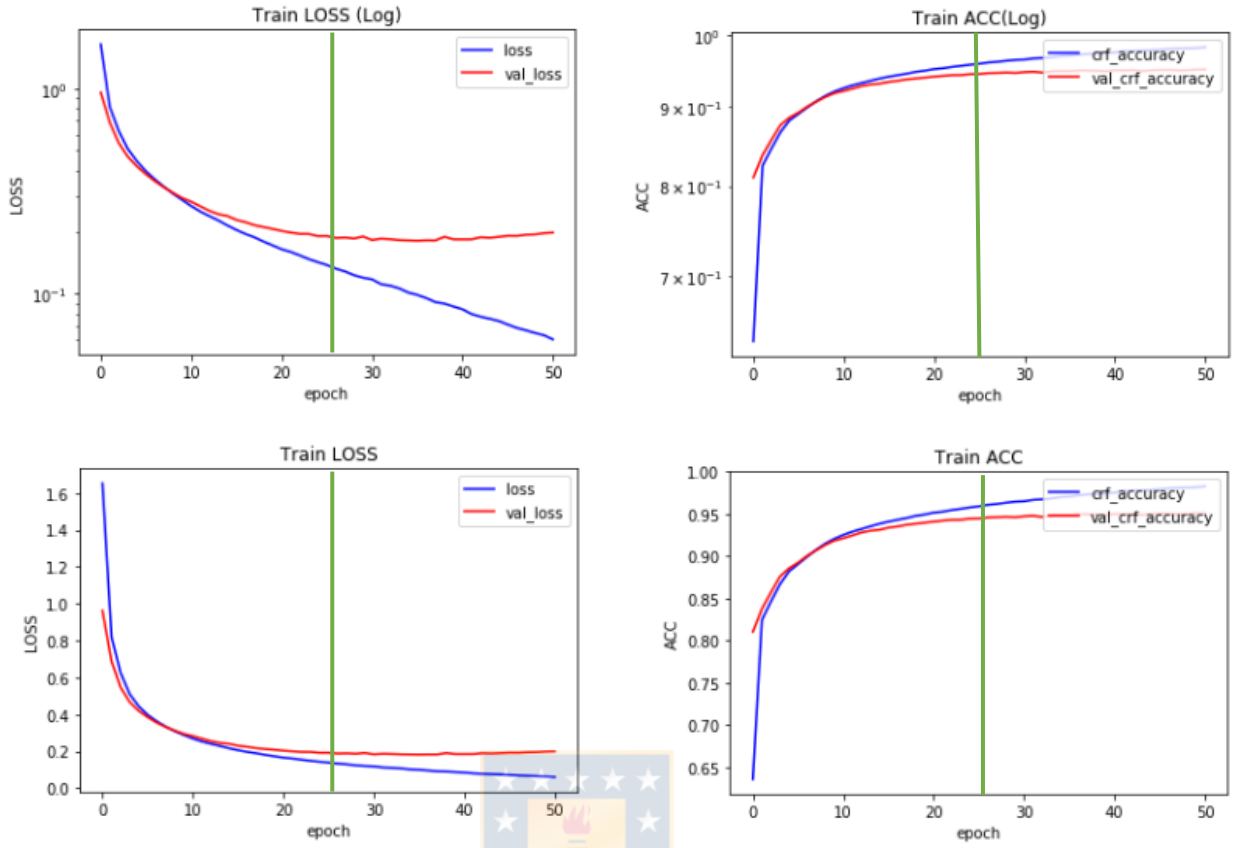


Fig. 6.2. Loss vs Epoch, Acc vs Epoch

Fuente: Elaboración propia.

Se entrenó a nivel de documento, con *batches* de 32 documentos. Como input de utilizó el *word-embedding* biomédico de 200 dimensiones entrenado por *skip*-gram propuesto por Jagannatha A. et al. (2016) [10]. Cada capa LSTM de la capa bidireccional es de 64 dimensiones. Se incluye *Batch Normalization* a la salida de la capa bidireccional. Tanto la capa *fully connecte*d como la capa CRF son de 19 dimensiones, considerando 18 clases etiquetadas y 1 de clase "None". Se determinó *early stopping* a las 25 épocas, identificando que, a partir de este punto, el modelo comienza a sobreajustarse al set de entrenamiento (Fig. 6.2).

B. Resultados del modelo entrenado Estado del Arte.

Se evalúan los resultados a nivel de palabras, utilizando *Accuracy, Precision, Recall y F1-Score.* Para evaluar el desempeño total se utiliza el promedio micro (promedio simple) y macro (promedio ponderado por el support de cada clase). A modo de permitir la evaluación sobre las clases de interés sin el impacto provocado por la clase "None" que corresponde a la clase predominante, se incluye el promedio micro y un promedio inverso, sin esas palabras. Se evalúa considerando cada

palabra en su clase original sin BIO-tagging (Fig. 6.3) y con BIO-tagging (Fig. 6.4). Se grafican los resultados de las métricas obtenidas mediante el reporte de clasificación (*Classification Report*) y la matriz de confusión normalizada, con el uso de la librería *Scikit-Learn*.

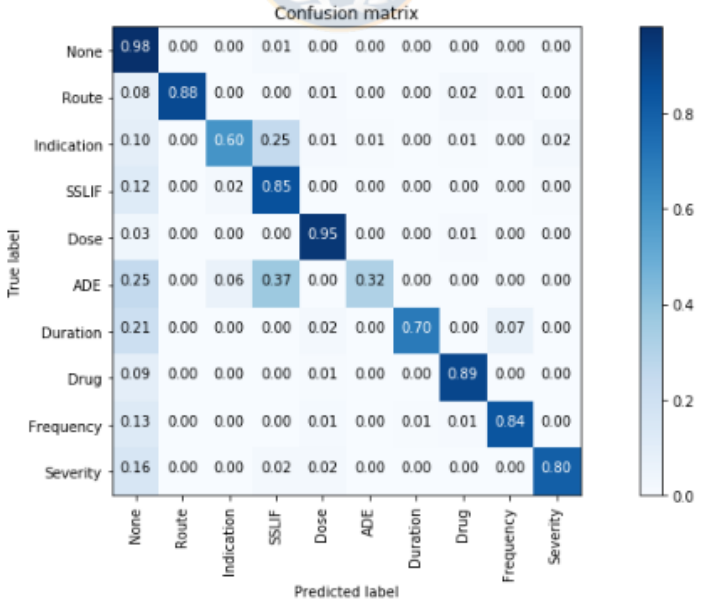
Accuracy:	0,95
Precision (micro)	0,95
Recall (micro)	0,95
F1-score (micro)	0,95
Precision (macro)	0,85
Recall (macro)	0,78
F1-score (macro)	0,81

Classification Report:

	Precision	Recall	F1-Score	Support
None	0,98	0,98	0,98	*97448*
Route	0,90	0,88	0,89	*382*
Indication	0,76	0,6	0,67	*1497*
SSLIF	0,81	0,85	0,83	*10075*
Dose	0,92	0,95	0,94	*1842*
ADE	0,75	0,32	0,44	*619*
Duration	0,74	0,7	0,72	*287*
Drug	0,95	0,89	0,92	*2895*
Frequency	0,91	0,84	0,87	*1850*
Severity	0,77	0,8	0,79	*730*

	Precision	Recall	F1-Score	Support
macro avg	0,85	0,78	0,81	*117625*
micro avg	0,95	0,95	0,95	*117625*
macro avg without None label:	0,83	0,76	0,79	*20177*
inverse avg without None label	0,80	0,72	0,75	*20177*

(a)



(b)

Fig. 6.3. Resultados sobre Test Set a nivel de etiqueta original.

(a) Resultados por reporte de clasificación. (b) Matriz de confusión normalizada.
Fuente: Elaboración propia.

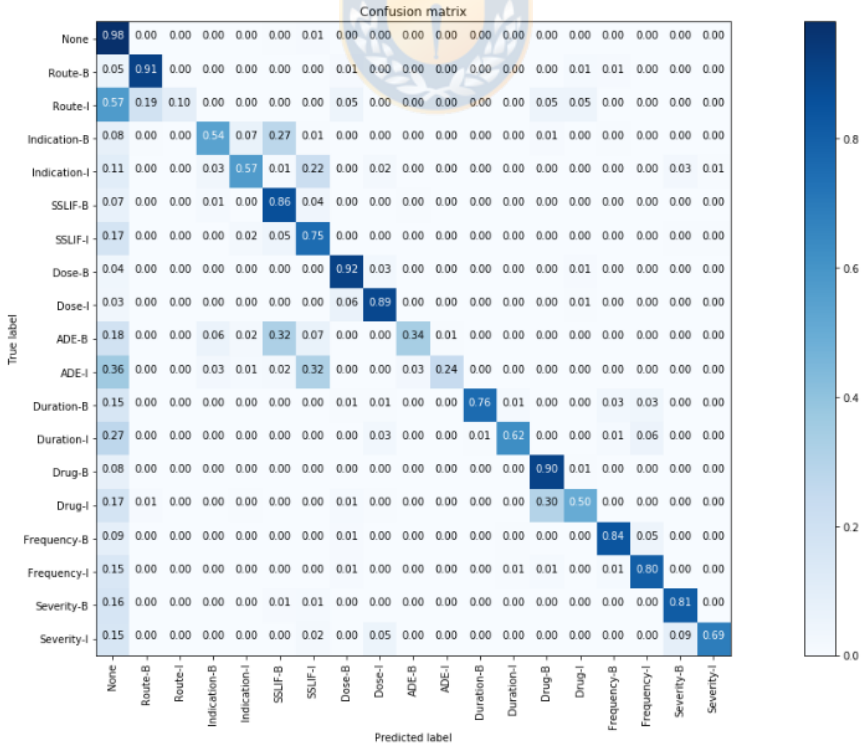
Accuracy:	0,95
Precision (micro)	0,95
Recall (micro)	0,95
F1-score (micro)	0,95
Precision (macro)	0,85
Recall (macro)	0,78
F1-score (macro)	0,81

Classification Report:

	Precision	Recall	F1-Score	Support
None	0,98	0,98	0,98	*97448*
Route-B	0,89	0,91	0,9	*361*
Route-I	0,67	0,1	0,17	*21*
Indication-B	0,71	0,54	0,61	*638*
Indication-I	0,70	0,57	0,63	*859*
SSLIF-B	0,83	0,86	0,85	*5333*
SSLIF-I	0,70	0,75	0,73	*4742*
Dose-B	0,85	0,92	0,88	*872*
Dose-I	0,90	0,89	0,89	*970*
ADE-B	0,74	0,34	0,47	*386*
ADE-I	0,67	0,24	0,35	*233*
Duration-B	0,82	0,76	0,79	*143*
Duration-I	0,65	0,62	0,64	*144*
Drug-B	0,91	0,9	0,91	*2446*
Drug-I	0,76	0,5	0,6	*449*
Frequency-B	0,89	0,84	0,87	*638*
Frequency-I	0,87	0,8	0,84	*1212*
Severity-B	0,73	0,81	0,77	*496*
Severity-I	0,78	0,69	0,73	*234*

	Precision	Recall	F1-Score	Support
macro avg	0,79	0,69	0,72	*117625*
micro avg	0,95	0,95	0,95	*117625*
macro avg without None label:	0,78	0,67	0,7	*20177*

(a)



(b)

Fig. 6.4. Resultados sobre Test Set a nivel de etiqueta utilizando BIO tagging.
(a) Resultados por reporte de clasificación. (b) Matriz de confusión normalizada.
Fuente: Elaboración propia.

C. Interpretación de resultados Estado del Arte.

En los resultados se observa claramente el efecto del desbalance de clases, siendo la clase "*None*" la que presenta la mejor *Precision*, *Recall* y *F1-Score*, sobre todas las clases de interés. Además, es posible identificar que la clase "*ADE*" presenta los peores resultados, con una mayor confusión con las clases "*Indication*" y "*SSLIF*", lo que significa que es etiquetada erróneamente como alguna de las otras dos entidades. Este problema puede explicarse debido a que las clases "ADE", "Indication" y "SSLIF" comparten vocabulario, debido a que, las tres clases corresponden a descripción de síntomas, ya sea como: efectos adversos, ser los activamente tratados o de otro tipo (por ejemplo, del historial clínico), respectivamente.

Al descomponer los resultados de las entidades etiquetadas mediante *BIO-tagging* (que corresponde a la forma en la que se entrenó el modelo), se puede observar el desempeño para identificar la primera palabra de una entidad, y las siguientes. Se observa, por ejemplo, una menor precisión para identificar *"Route-I"*, la cual se confunde con *"Route-B",* principalmente, debido a que la mayoría de las entidades de este tipo solo tienen una palabra. Sin embargo, esta confusión intra-clase, no perjudica al desempeño para determinar que la palabra pertenece a la entidad *"Route".* Algo similar es lo que ocurre con la entidad *"Drug".* Por otro lado, entidades como "*Dose*", "*Duration*", "*Frequency*" y "*Severity*", poseen una menor confusión entre sus sub-entidades "*-B*" e "*-I*". Sin embargo, se observa confusión con la entidad "*None*" de prácticamente todas las entidades que corresponden a frases y, por lo tanto, son de mayor extensión, mayor vocabulario y "*stopwords*" dentro de la entidad.

Las observaciones sobre los resultados del modelo replicado se condicen con los presentados en el Estado del Arte [9] [10]. Sin embargo, este no señala los documentos específicos que se utilizaron para el entrenamiento y evaluación, presenta sus resultados a nivel de entidad (considerando la totalidad de la expresión) y no incluye los resultados sobre las palabras sin etiquetas, cuyo efecto sobre los resultados se decidió explorar en la presente investigación.

6.2. Evaluación de algoritmo de clustering.

Se utilizó *K-means* como algoritmo de clústering, sobre el espacio de veinte dimensiones entregado para cada palabra, en la capa *fully-connected* del modelo entrenado. Este arreglo de datos representaría las características utilizadas por la capa CRF para la clasificación, además de resumir información de la palabra etiquetada con su contexto, extraído a partir de la capa LSTM bidireccional. Se utilizó k = 42, de acuerdo a lo descrito en la metodología. Se realizó una inspección visual mediante

T-SNE (del inglés, t-distributed Stochastic Neighbor Embedding) [71] que es un método de reducción de dimensionalidad para visualizar datos. Se evaluó la pertenencia de las clases originales a las nuevas agrupaciones mediante un mapa de calor y se revisó el vocabulario agrupado mediante nubes de palabras.

A. Visualización mediante T-SNE.

Con el fin de realizar una inspección visual del clústering mediante T-SNE [71], se selecciona un 10% de las palabras del set de validación mediante muestreo aleatorio estratificado de acuerdo a las entidades originales etiquetadas por expertos. Se normalizan las variables utilizadas para el entrenamiento del algoritmo *k-means*, y se aplica *T-SNE* a dos dimensiones. En Fig. 6.5 se visualizan las entidades originales, observando que la mayor parte del corpus corresponde a la entidad "*None*". Se observa que las caracteristicas entrenadas por el modelo de clasificación logra agrupar en su mayoría a las clases de interés. Sin embargo, es posible visualizar la confusión de clases "*Indication*", "*SSLIF*" y "*ADE*". Mientras que clases como "*Drug*", "*Dose*" y "*Route*" se ven mucho más definidas.

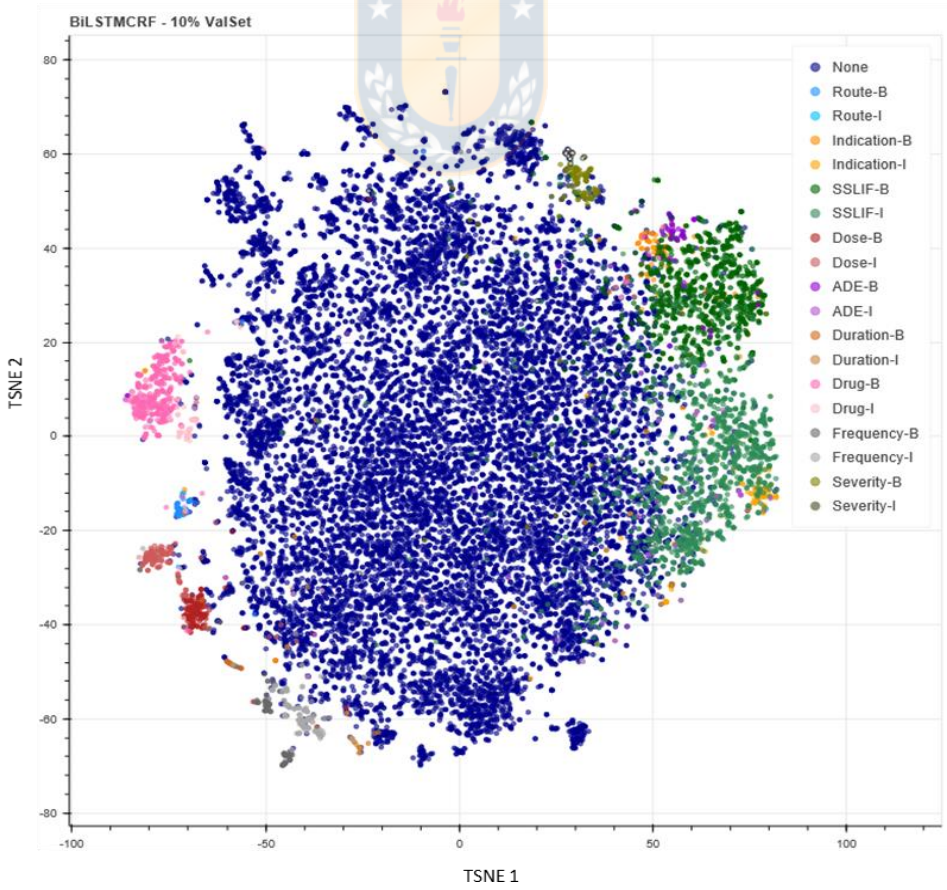


Fig. 6.5. Entidades originales sobre el corpus de validación.
Fuente: Elaboración propia.

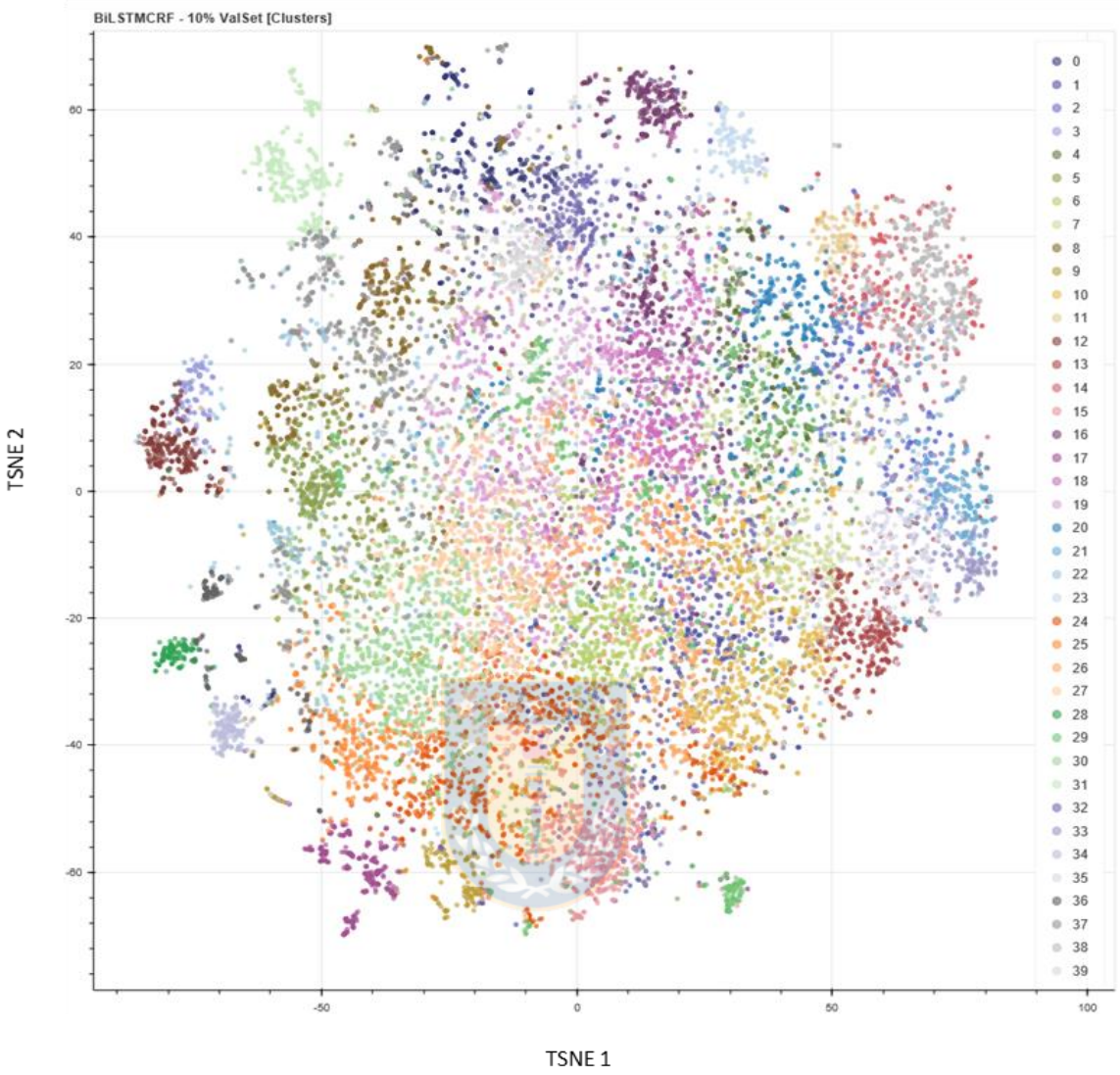


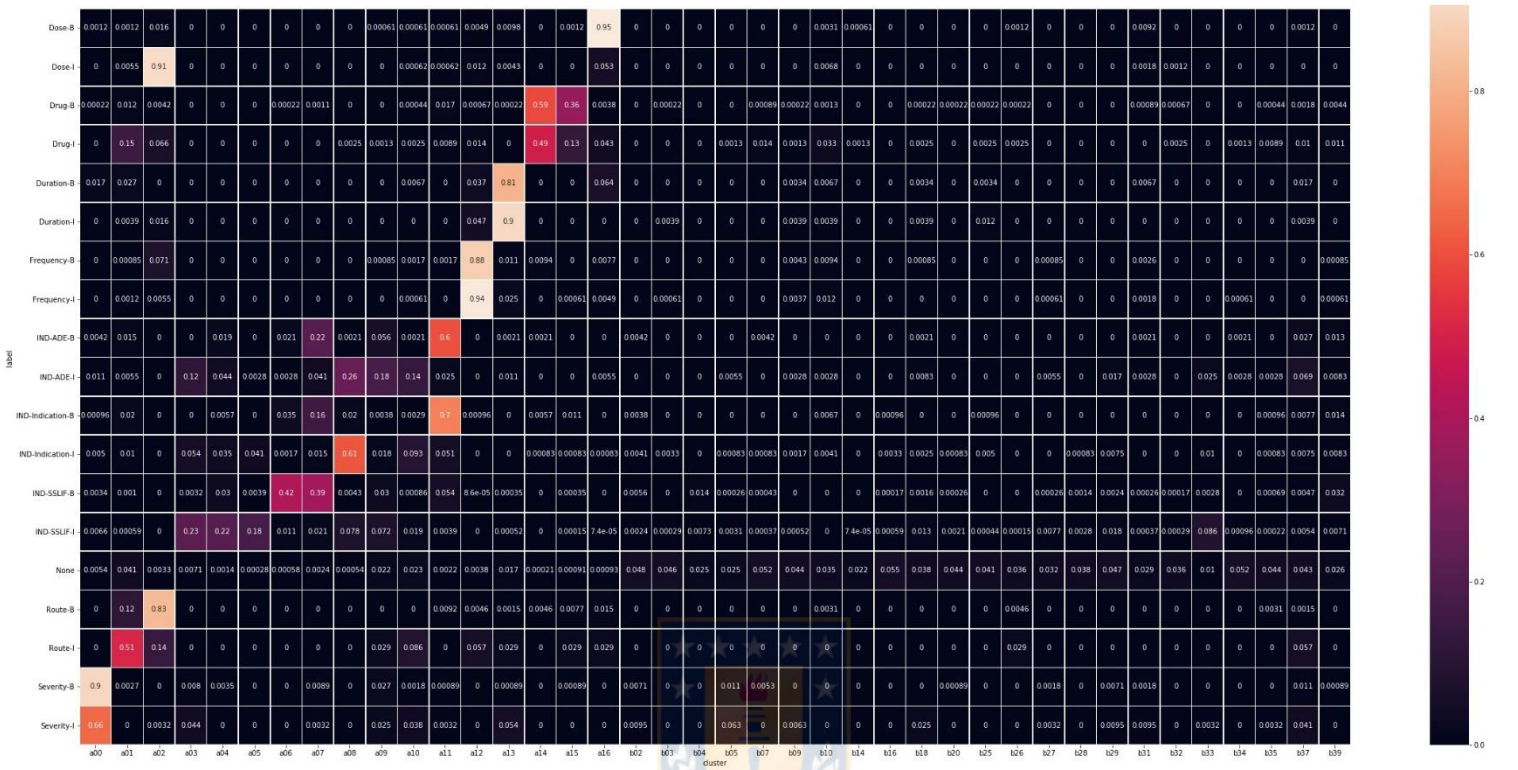
Fig. 6.6. Clústeres determinados por *K-means* sobre el corpus de validación.
Fuente: Elaboración propia.

En Fig. 6.6, se visualizan las agrupaciones determinadas mediante *k-means* sobre la misma muestra que se visualiza en Fig. 6.5. Si bien, debido a la cantidad de agrupaciones, no es posible distinguir claramente cada clúster, es posible identificar la subdivisión de algunas de las entidades originales en nuevos clústers, pero, principalmente, se observa la subdivisión sobre la clase "None", que ahora corresponde a múltiples agrupaciones.

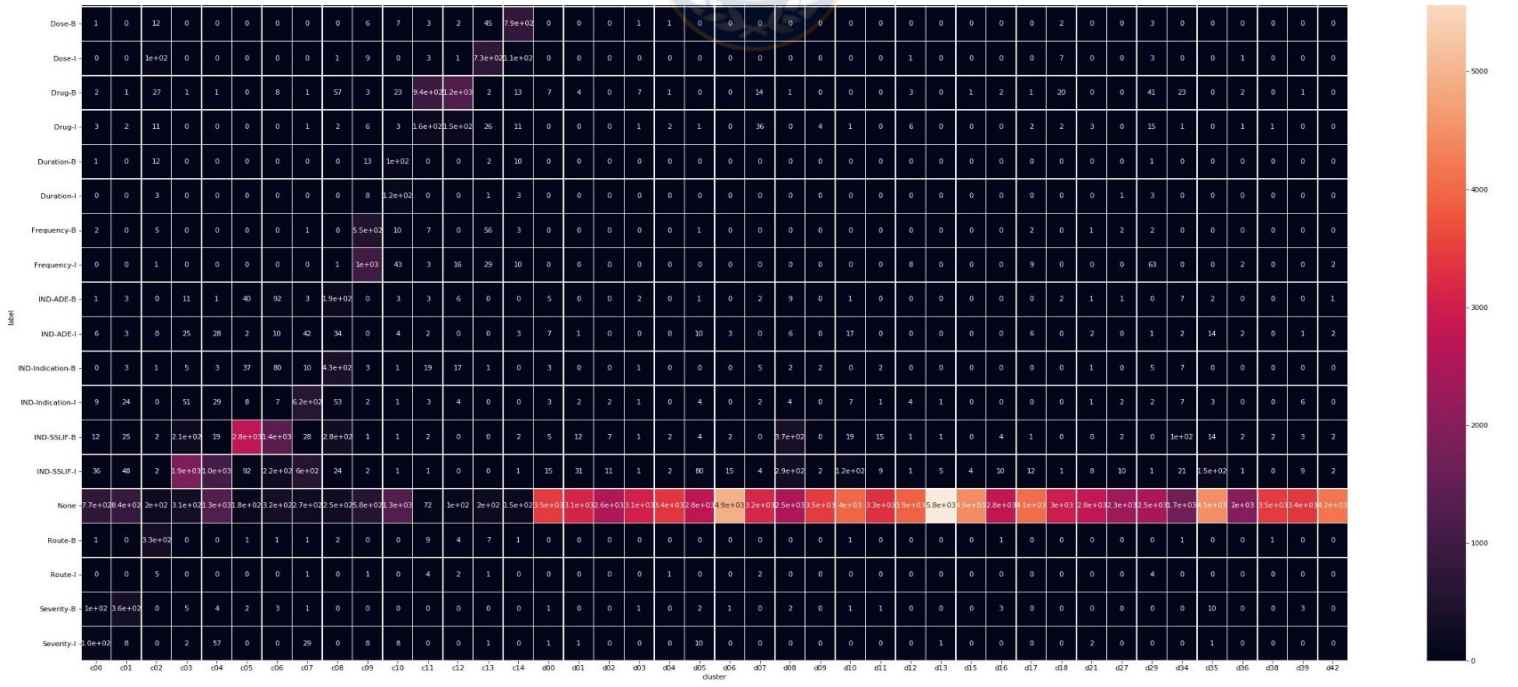
B. Pertenencia de clases a cada clúster.

En la Fig. 6.7.a, se presenta el mapa de calor normalizado del total de palabras del conjunto de validación de clúster vs clase original. En la Fig. 6.7.b se visualiza el mapa de calor sin normalizar. Se les asignó nombre a las agrupaciones con el fin de facilitar la visualización; considerando "*aXX*"

a las agrupaciones con un mayor porcentaje de pertenencia de palabras etiquetadas y "*bXX*" a las agrupaciones con un mayor porcentaje de palabras no etiquetadas ("*None*").



(a)



(b)

Fig. 6.7. Mapa de calor de pertenencia de cluster vs entidad original.
(a) Normalizado (% de clase original). (b) No normalizado (número de palabras).
Fuente: Elaboración propia.

Al analizar la pertenencia se observa que las clases de interés se tienden a agrupar en 17 clústeres, sin embargo, ninguno logra mejorar la precisión con respecto al algoritmo de clasificación, pues se encuentra sin la participación de la capa de clasificación CRF del modelo supervisado. Es interesante observar que la mayor parte del corpus, que correspondía al texto sin etiquetar, se agrupa principalmente en los clústeres restantes, lo que podría contribuir a balancear la tarea supervisada.

C. Nube de palabras por clúster.

Se visualiza la nube de palabras formada por cada clúster.

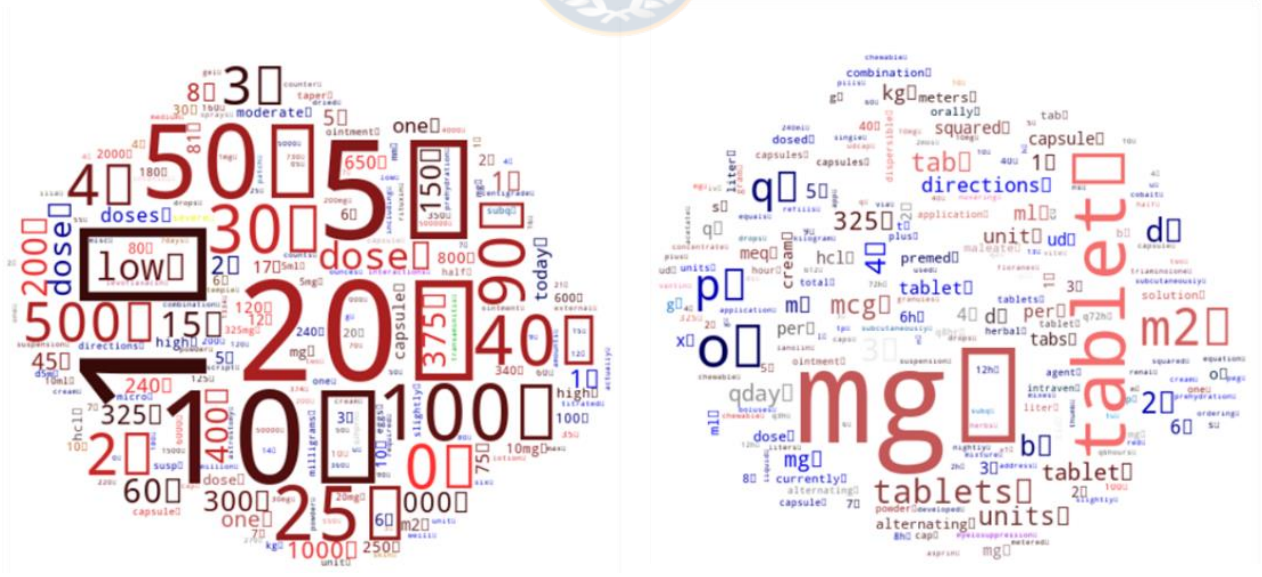
(a)

(b)

Fig. 6.8. Agrupaciones con mayor pertenencia de entidad “Drug”.
(a) Clúster a13. (b) Clúster a14. /Fuente: Elaboración propia.

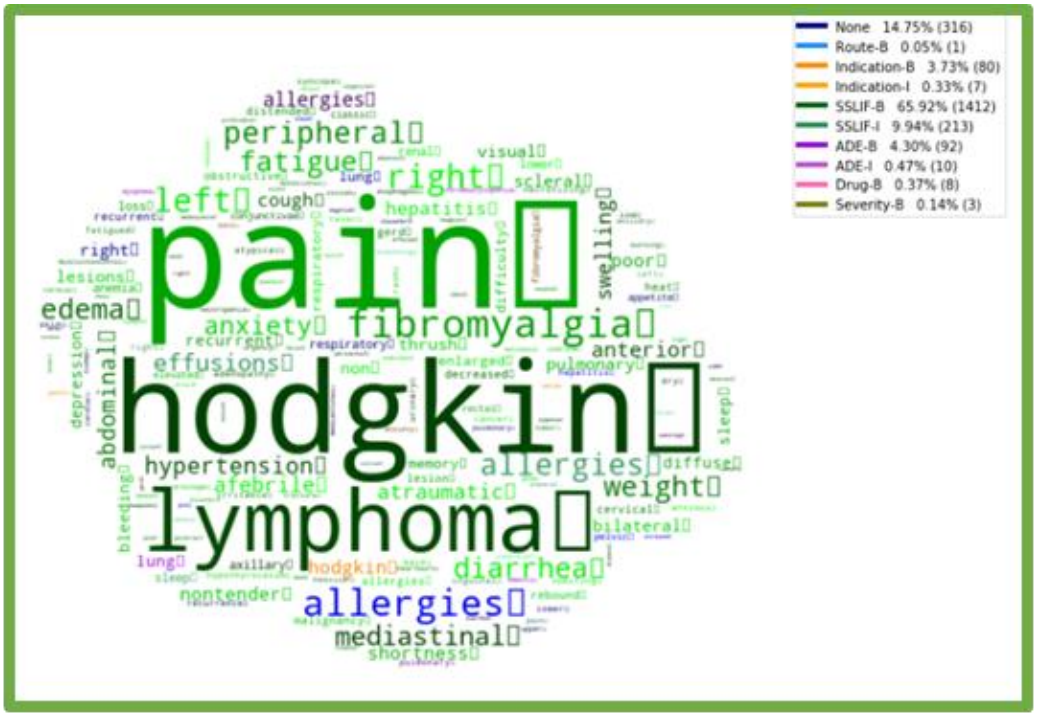
Se identifica que algunas entidades que se subdividen, como, por ejemplo, la entidad “Drug” que corresponde a drogas y medicamentos, se subdivide en su mayoría a los clústeres a13 y a14 (Fig. 6.8). Mientras en el primer clúster se observa que la palabra de mayor frecuencia es “chemotherapy”, en el segundo lo es la “prednisona”. En una exploración visual de ambas agrupaciones, es interesante ver que la mayoría de los medicamentos del primer grupo corresponden a principios activos o nombres comerciales de drogas asociados a quimioterapia, mientras que en la segunda agrupación se observan drogas asociadas a tratamiento de efectos secundarios o de apoyo, como antiinflamatorios o anticoagulantes.

Otras agrupaciones son capaces de identificar las etiquetas de BIO-tagging asignadas para el entrenamiento. Como ocurre en el caso de la entidad “Dose” (Fig. 6.9) que corresponde a la dosis asociada a una droga o medicamento, generalmente, corresponde a dos palabras: a un número (“*Dose-B*”) y su respectiva unidad de medida (“*Dose-I*”). La unidad de medida es expresada como una abreviación (“mg”, “ml”, “tab”, etc.) por lo que este clúster además agrupa a la entidad “Route”, que corresponde a la ruta de administración del medicamento, que también es expresada como abreviación (“p.o”, “p”, “o”, etc.).

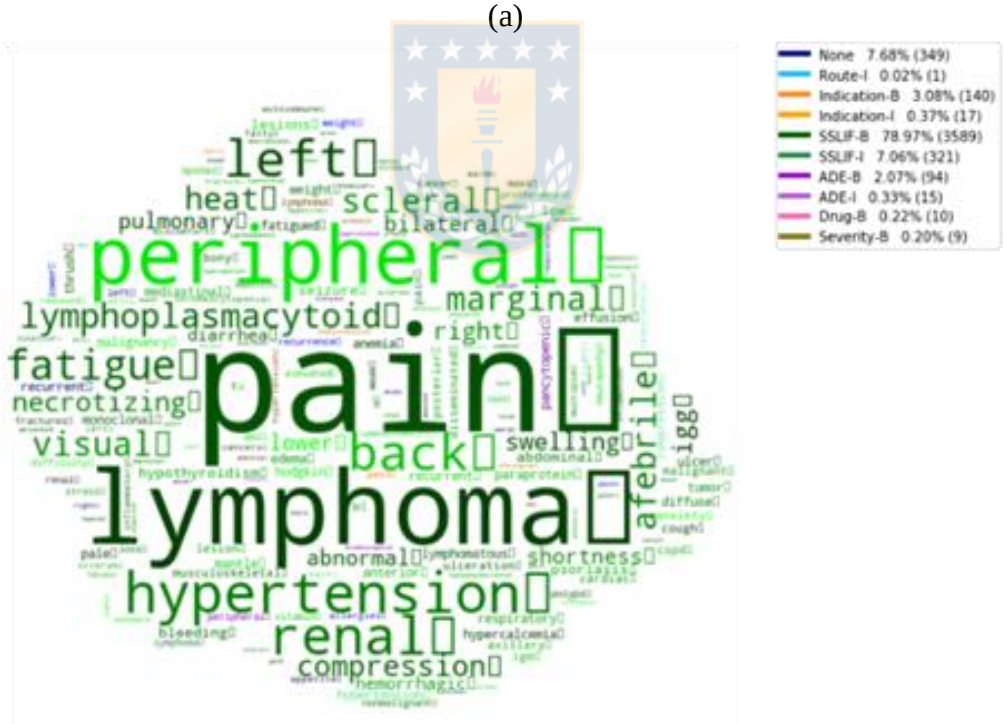


(a) (b)

Fig. 6.9. Agrupaciones con mayor pertenencia de entidad “Dose”.
(a) Clúster a16 “Dose-B” (b) Clúster a02. “Dose-I”.
Fuente: Elaboración propia.



(a)

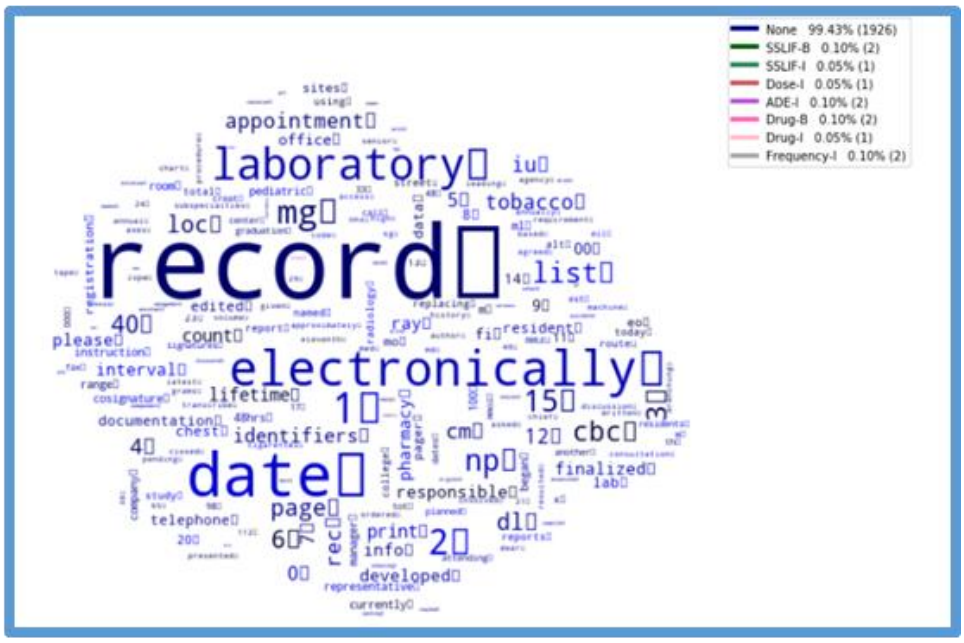


(b)

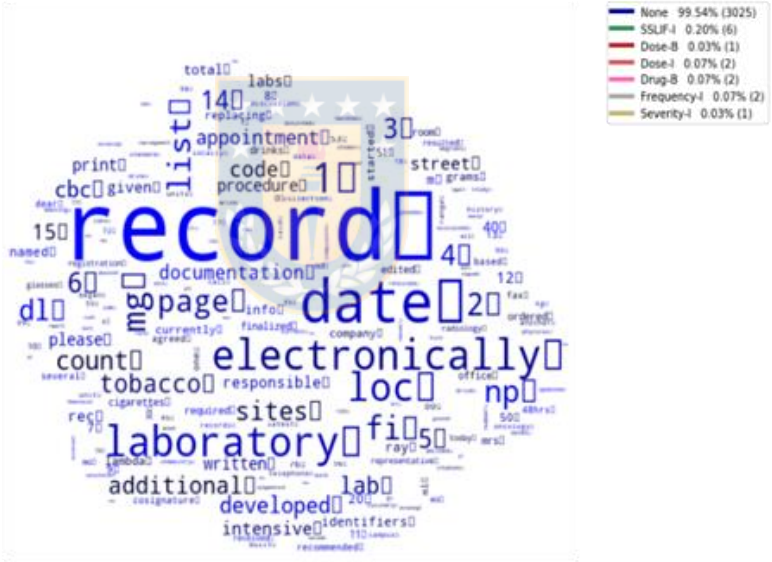
Fig. 6.10. Nube de palabras en cluster mayoritariamente de entidad "SSLIF"

(a) Clúster sobre set de entrenamiento (b) Clúster sobre set de validación (Fuente: Elaboración propia)

Como se puede observar en la Fig. 6.10. el clúster presenta presencia de palabras etiquetadas como "SSLIF", repitiéndose entre las más frecuentes "*pain*" y "*lymphoma*", tanto para el set de entrenamiento como validación.



(a)



(b)

Fig. 6.11. Nube de palabras en cluster mayoritariamente "None"

(a) Set de entrenamiento (b) Set de validación (Fuente: Elaboración propia)

En la se presenta una de las agrupaciones formada principalmente por términos no etiquetados, donde se repiten entre las palabras más frecuentes: "*record*", "*electronically*", "*date*" y "*laboratory*", Estas palabras podrían estar asociadas a la descripción del registro médico electrónico y podrían representar una nueva clase que se podría anotar mediante la supervisión de expertos. Esta tendencia puede ser explorada en profundidad, en la comparación de nube de palabras para los 42 clústeres formados, donde se presenta la agrupación formada para los documentos de entrenamiento y validación, ordenados por presencia de las clases de interés, disponible en el Anexo B.

6.3. Resultados sobre corpus re-etiqueado

A. Entrenamiento.

Se entrena el modelo del Estado de Arte, para el corpus re-etiquetado. Para este re-etiquetado se consideran los 42 clústeres definidos previamente, y se le asigna como nueva etiqueta a las palabras que no fueron etiquetadas por expertos. En la exploración de los clústeres se identificaron 15 clústeres que tendían a confundirse con las clases originales. Se agruparon en "c-IND", los clústeres que solían confundirse con las clases de indicación médica ("Indication", "SSLIF", "ADE"), y se agruparon en "c-Drug" el conjunto de clústeres que se confundían con la etiqueta "Drug". Se reduce a 36 etiquetas por clústeres junto a 18 etiquetas originales. Se entrenó a nivel de documento, con batches de 32 documentos. Cada capa LSTM de la capa bidireccional es de 64 dimensiones. Se incluye Batch Normalization a la salida de la capa bidireccional. Se determinó early stopping a las 40 épocas, identificando que, a partir de este punto, el modelo comienza a sobre-ajustarse al set de entrenamiento (Fig. 6.12).

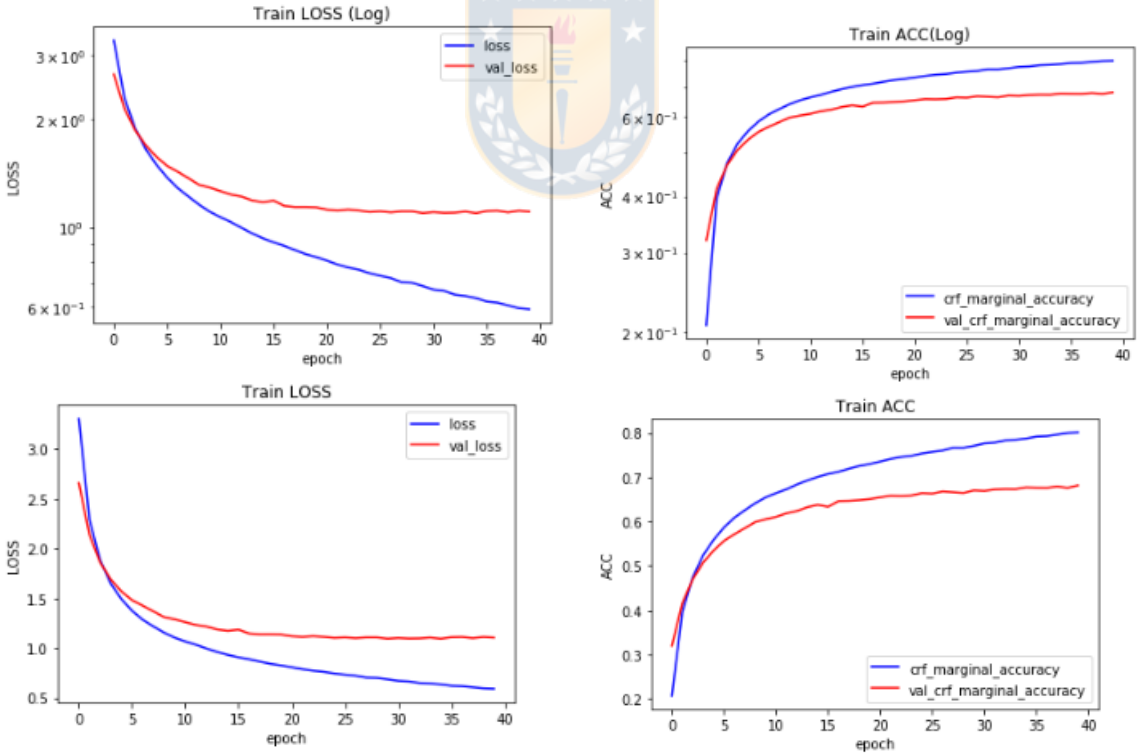


Fig. 6.12. Loss vs Epoch, Acc vs Epoch (Conjunto de Entrenamiento y Validación)
Fuente: Elaboración propia.

B. Resultados.

Se presentan los resultados a través del reporte de clasificación y la matriz de confusión normalizada. En primer lugar, analizando el detalle por cada clase original a través de BIO-Tagging (Fig. 6.13 y Fig. 6.14). Para mejorar la visualización se ordenan en primer lugar las 18 etiquetas originales (XX-B y XX-I), luego las 9 etiquetas de clústeres que se confunden con las clases originales (c-XX) y finalmente los 27 clústeres restantes (d-XX). En la Fig. 6.15, se visualizan los resultados para las clases originales de la tarea NER, 9 clases etiquetas por expertos y la clase "None".

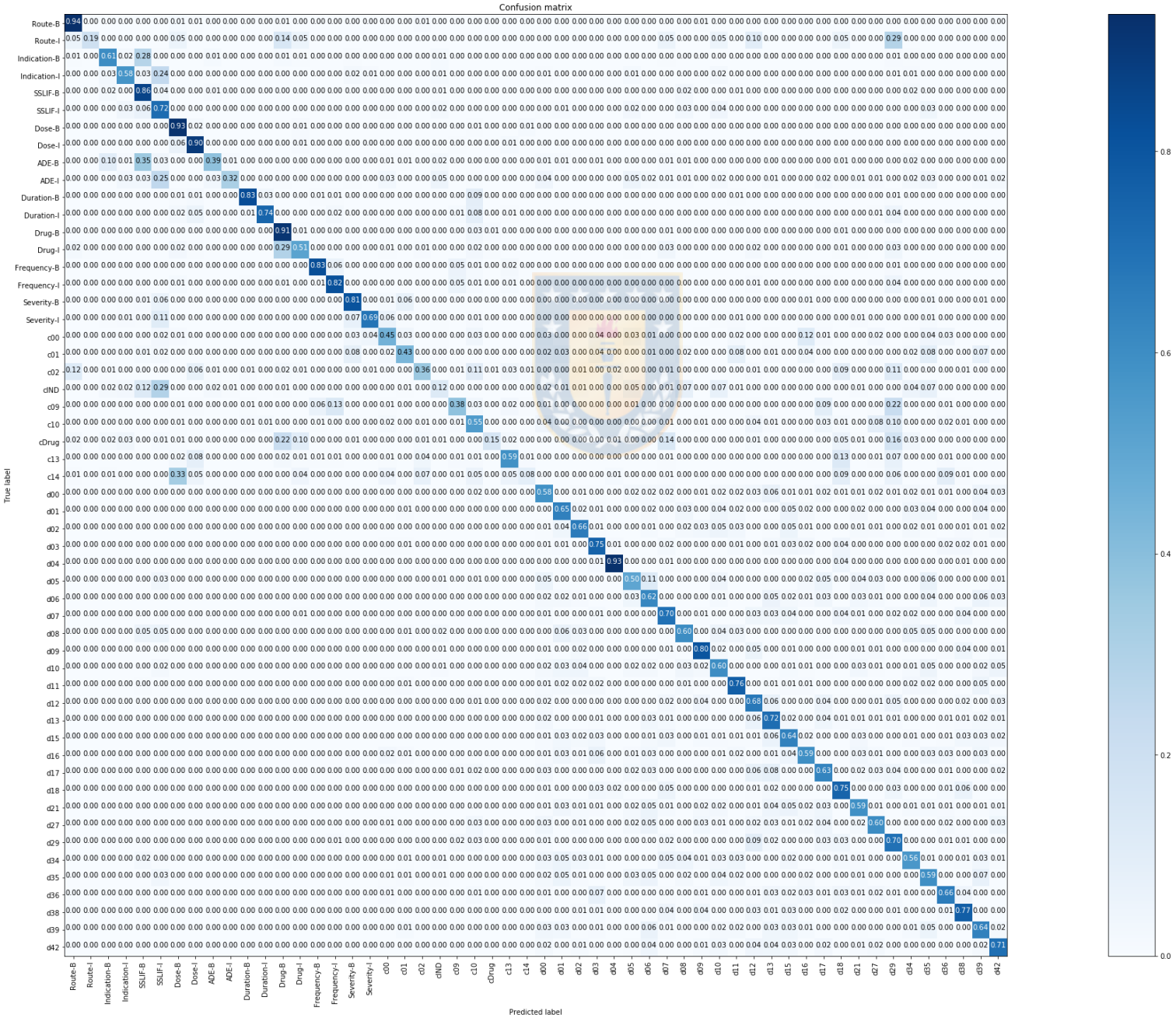


Fig. 6.13. Matriz de Confusion Normalizado.
Nota: modelo re-etiquetado sobre Test Set (detalle por etiqueta). Fuente: Elaboración propia.

Accuracy:	0,68
Precision (micro)	0,68
Recall (micro)	0,68
F1-score (micro)	0,68
Precision (macro)	0,65
Recall (macro)	0,62
F1-score (macro)	0,63

Classification Report:

	Precision	Recall	F1-Score	Support
Route-B	0,86	0,94	0,89	*361*
Route-I	0,40	0,19	0,26	*21*
Indication-B	0,66	0,61	0,63	*638*
Indication-I	0,66	0,58	0,62	*859*
SSLIF-B	0,81	0,86	0,84	*5333*
SSLIF-I	0,69	0,72	0,7	*4742*
Dose-B	0,82	0,93	0,87	*872*
Dose-I	0,92	0,9	0,91	*970*
ADE-B	0,52	0,39	0,44	*386*
ADE-I	0,61	0,32	0,42	*233*
Duration-B	0,79	0,83	0,81	*143*
Duration-I	0,74	0,74	0,74	*144*
Drug-B	0,90	0,91	0,91	*2446*
Drug-I	0,68	0,51	0,58	*449*
Frequency-B	0,90	0,83	0,86	*638*
Frequency-I	0,87	0,82	0,84	*1212*
Severity-B	0,71	0,81	0,76	*496*
Severity-I	0,70	0,69	0,7	*234*
c00	0,52	0,45	0,48	*681*
c01	0,50	0,43	0,46	*724*
c02	0,55	0,36	0,43	*190*
cIND	0,41	0,12	0,18	*2008*
c09	0,49	0,38	0,43	*499*
c10	0,56	0,55	0,55	*1371*
cDrug	0,38	0,15	0,22	*166*
c13	0,56	0,59	0,57	*198*
c14	0,28	0,08	0,13	*154*
d00	0,56	0,58	0,57	*3109*
d01	0,62	0,65	0,63	*3198*
d02	0,65	0,66	0,66	*2644*
d03	0,73	0,75	0,74	*3746*
d04	0,95	0,93	0,94	*3713*
d05	0,54	0,5	0,52	*1927*
d06	0,59	0,62	0,6	*3879*
d07	0,65	0,7	0,68	*3323*
d08	0,60	0,6	0,6	*2123*
d09	0,78	0,8	0,79	*3723*
d10	0,59	0,6	0,59	*3439*
d11	0,78	0,76	0,77	*3529*
d12	0,63	0,68	0,65	*4437*
d13	0,67	0,72	0,69	*6273*
d15	0,66	0,64	0,65	*5114*
d16	0,67	0,59	0,63	*2670*
d17	0,63	0,63	0,63	*3504*
d18	0,74	0,75	0,74	*3664*
d21	0,58	0,59	0,59	*2668*
d27	0,63	0,6	0,61	*2212*
d29	0,63	0,7	0,66	*2753*
d34	0,51	0,56	0,54	*1821*
d35	0,55	0,59	0,57	*3150*
d36	0,72	0,66	0,69	*2337*
d38	0,77	0,77	0,77	*4775*
d39	0,60	0,64	0,62	*3493*
d42	0,72	0,71	0,71	*4233*

	Precision	Recall	F1-Score	Support
macro avg	0,65	0,62	0,63	*117625*
micro avg	0,68	0,68	0,68	*117625*

Fig. 6.14. Reporte de Clasificación.

Nota: modelo re-etiquetado sobre Test Set (detalle por etiqueta). Fuente: Elaboración propia.

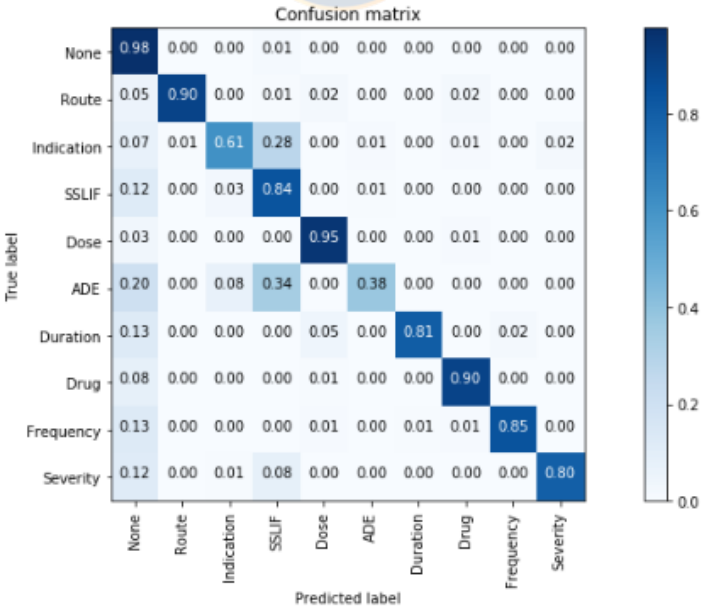
Accuracy:	0,95
Precision (micro)	0,95
Recall (micro)	0,95
F1-score (micro)	0,95
Precision (macro)	0,81
Recall (macro)	0,80
F1-score (macro)	0,81

Classification Report:

	Precision	Recall	F1-Score	Support
None	0,98	0,98	0,98	*97448*
Route	0,85	0,9	0,88	*382*
Indication	0,68	0,61	0,65	*1497*
SSLIF	0,80	0,84	0,82	*10075*
Dose	0,91	0,95	0,93	*1842*
ADE	0,57	0,38	0,48	*619*
Duration	0,79	0,81	0,80	*287*
Drug	0,92	0,9	0,91	*2895*
Frequency	0,90	0,85	0,88	*1850*
Severity	0,73	0,8	0,77	*730*

	Precision	Recall	F1-Score	Support
macro avg	0,81	0,80	0,81	*117625*
micro avg	0,95	0,95	0,95	*117625*
macro avg without None label:	0,80	0,78	0,79	*20177*
inverse avg without None label:	0,77	0,77	0,77	*20177*

(a)



(b)

Fig. 6.15. Resultados modelo re-etiquetado sobre Test Set a nivel de etiqueta original.
(a) Resultados por reporte de clasificación. (b) Matriz de confusión normalizada.
Fuente: Elaboración propia.

6.4. Comparación de resultados.

Se realiza una comparación de los resultados de los métodos descritos anteriormente en TABLA 6.1 y TABLA 6.2.

TABLA 6.1. RESULTADOS POR ENTIDAD.

		Modelo del Estado-del-Arte			Modelo Re-Etiquetado		
Entidad	**Support**	**Pr**	**Re**	**F1**	**Pr**	**Re**	**F1**
None	97448	0,98	0,98	0,98	0,98	0,98	0,98
Route	382	**0,90**	0,88	0,89	0,85	**0,90**	0,88
Indication	1497	0,76	0,60	0,67	**0,68**	**0,61**	0,65
SSLIF	10075	**0,81**	**0,85**	0,83	0,80	0,84	0,82
Dose	1842	**0,92**	**0,95**	0,94	0,91	**0,95**	0,93
ADE	619	**0,75**	0,32	0,44	0,57	**0,38**	**0,48**
Duration	287	0,74	0,70	0,72	**0,79**	**0,81**	**0,80**
Drug	2895	**0,95**	0,89	0,92	0,92	**0,90**	0,91
Frequency	1850	**0,91**	0,84	0,87	0,90	**0,85**	**0,88**
Severity	730	**0,77**	**0,80**	0,79	0,73	**0,80**	0,77

Las métricas son descritas en Capítulo 2, sección 2.4.

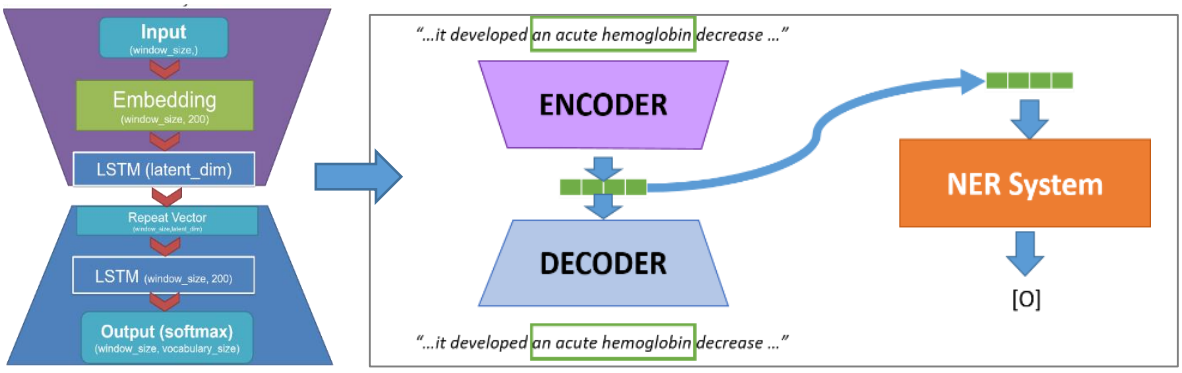
TABLA 6.2. RESULTADO PROMEDIO DE LA TAREA NER.

		Modelo del Estado-del-Arte			Modelo Re-Etiquetado		
Promedio	**Support**	**Pr**	**Re**	**F1**	**Pr**	**Re**	**F1**
Macro.	117625	**0,85**	0,78	0,81	0,81	**0,80**	0,81
Micro.	117625	0,95	0,95	0,95	0,95	0,95	0,95
Macro (sin None)	20177	**0,83**	0,76	0,79	0,80	**0,78**	0,79
Inverso (Sin None)	20177	**0,80**	0,72	0,75	0,77	**0,77**	**0,77**

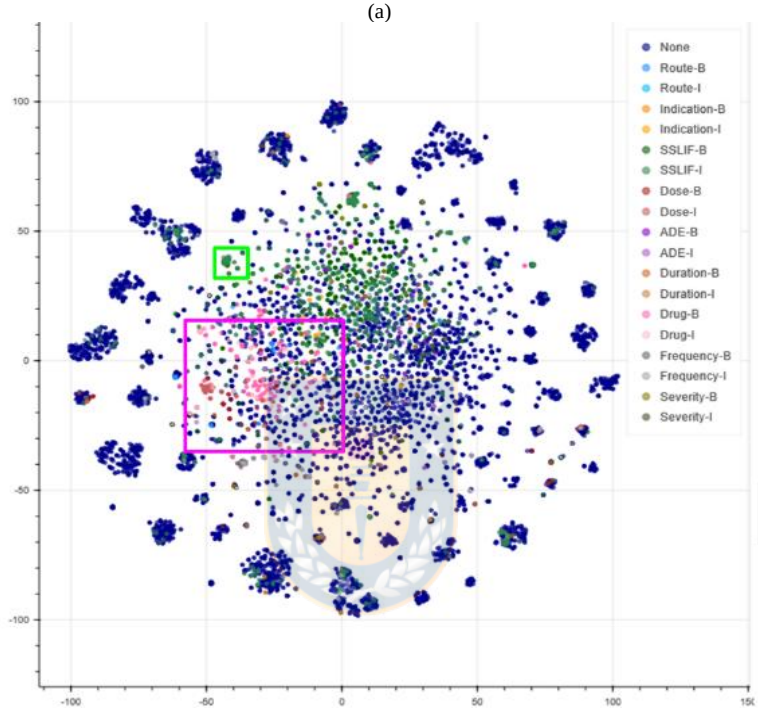
Las métricas son descritas en Capítulo 2, sección 2.4.

6.5. Aproximaciones descartadas.

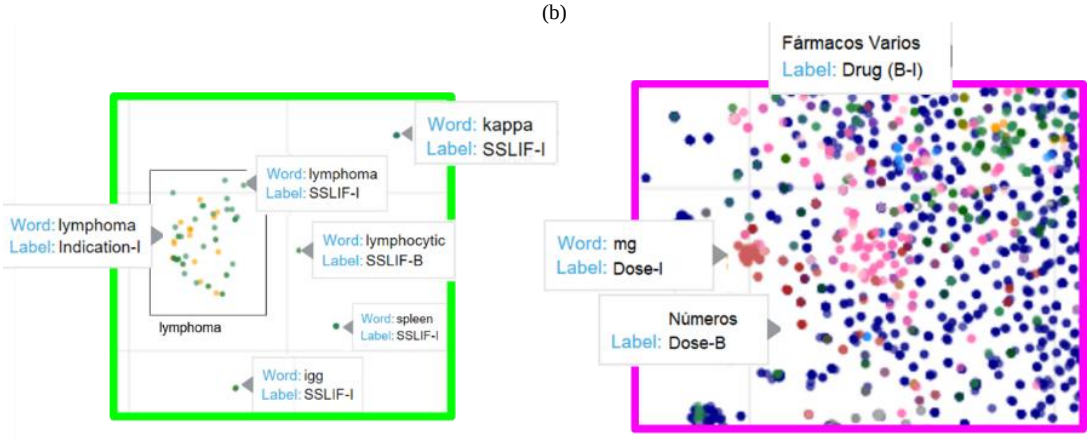
A continuación, se describirán de forma superficial, algunas de las primeras aproximaciones de modelos propuestos para la tarea NER den el marco del presente trabajo de investigación, pero que fueron descartados. Una de las primeras aproximaciones propuestas en la exploración de mejoras para el sistema de reconocimiento de entidades, buscaba el uso de un sistema enriquecido por características aprendidas por un *autoencoder* (Fig. 6.16.a) que lograra codificar secuencias de texto de diferente extensión. Se utilizó una representación de ventana variable de contexto, para representar las secuencias de palabras, y luego utilizar el espacio latente del *autoencoder* para determinar la entidad de cada palabra mediante un clasificador. Al explorar el espacio latente (Fig. 6.16.b), se observó que el *autoencoder* facilitaba una reducción de la dimensionalidad del *Word-embedding*, agrupando las palabras por su valor semántico individual y no por su contexto (Fig. 6.16.c).



(a)



(b)



(c) (d)

Fig. 6.16. ***Visualización del espacio latente entrenado.***

(a) Modelo Autoencoder propuesto. (b) Espacio latente entrenado. (c) Clúster Lymphoma. (d) Términos asociados

Fuente: Elaboración propia.

Como segunda instancia, se propusieron variaciones al autoencoder, mediante el entrenamiento mediante la predicción de la continuación de una ventana de texto, para usar estas características para la tarea NER. Se probaron ventanas de 3, 5 y 7 palabras, que se denominaron "enc-3", "enc-5" y "enc-7", respectivamente. Sin embargo, el entrenamiento de esta tarea no resultó ser tan eficiente, debido a la alta variabilidad del texto, además de no generar características que presentaran una mejora a los resultados. Finalmente, el uso de estas características presentó un desempeño peor al del estado del arte, aun probando con otros clasificadores, como *Random Forest (RF)*.

TABLA 6.3. RESULTADOS PARA LA TAREA NER

CLASIFICADOR	CARACTERÍSTICAS	PR	RE	F1-SCORE
RF	**enc-3**	**0.84**	**0.68**	**0.75**
RF	enc-5	0.84	0.57	0.67
RF	enc-7	0.81	0.52	0.62

Otra de las aproximaciones que se propuso, fue modificar directamente la función de costo del modelo de *Deep Learning* propuesto por el estado del arte, para premiar correctamente el desempeño sobre las clases de interés, considerando el desbalance de clases. Sin embargo, las implementaciones propuestas, empeoraron el entrenamiento del modelo, lo que se vio finalmente como peores resultados de clasificación.

6.6. Evaluación de resultados

En resumen, en el presente capítulo se presentaron los resultados de las diferentes etapas de la metodología propuesta En resumen, en el presente capítulo se presentaron los resultados de las diferentes etapas de la metodología propuesta. En primer lugar, se entrenó un modelo de reconocimiento de entidades, a nivel de palabra, para las 10 etiquetas originales del corpus. Esto, a través de un modelo de Bi-LSTM-CRF correspondiente al Estado del Arte, para el cual las etiquetas fueron separadas mediante BIO Tagging, generándose un total de 19 clases a entrenar, donde las 9 entidades de interés son separadas en B- e I- (18 clases) y se incluye la entidad O (palabras sin etiqueta). El modelo recibe un documento completo de anotación médica, y clasifica cada palabra de la secuencia en alguna de estas 19 clases obtenidas mediante BIO Tagging. Obteniendo como resultados (Fig. 6.4.a): Pr: 78%, Re: 67% y F1-score: 70%, como promedio macro de la tarea, sin incluir la entidad O. y: Pr: 79%, Re: 69% y F1-score: 72%, como promedio macro de la tarea,

incluyendo a las palabras no etiquetadas. No se entrega el promedio-micro, ya que se ve distorsionado debido al desbalance de clases.

Sin embargo, no es de mayor interés la confusión generada entre que una palabra sea etiquetada como B- o I-, más bien, si fue identificada como la entidad principal. Por esto, se presentan los resultados del modelo a nivel de etiqueta original (Fig. 6.3.a), obtenido: Pr: 83%, Re: 76% y F1-score: 79%, como promedio macro de la tarea, sin incluir la entidad O. y: Pr: 85%, Re: 78% y F1-score: 81%, como promedio macro de la tarea, incluyendo a las palabras no etiquetadas.

Es posible evaluar, el impacto de la mejora sobre las clases con menor presencia en el corpus, entregándoles a cada métrica una ponderación inversa a su proporción de presencia en el corpus. Llamaremos a esta métrica promedio macro inverso, siendo los resultados sobre la tarea original: Pr: 80%, Re: 72% y F1-score: 75% para las clases de interés.

Es interesante revisar el desempeño sobre las clases ADE, Indication y SSLIF. Como se explicó en capítulos previos, estas tres entidades comparten vocabulario y su clase tiene mayor relación al contexto, lo que facilita la confusión del modelo. Para la etiqueta ADE, un 25% de las palabras son confundidas como sin etiqueta (O), un 37% como SSLIF y solo un 32% correctamente como ADE. Para la etiqueta Indication, un 10% de las palabras son confundidas como sin etiqueta (O), un 25% como SSLIF y un 60% es etiquetado correctamente. En ambas etiquetas se observa una confusión debido al desbalance de clases hacia las etiquetas O y SSLIF, que corresponden a las de mayor presencia en el corpus. Esta mayor presencia facilita su reforzamiento en el entrenamiento y mejor desempeño, siendo que para “O” un 98% de las palabras de esta clase son etiquetadas correctamente y para SSLIF un 85%.

Posteriormente, se entrenó un algoritmo de clústering de aprendizaje no supervisado, sobre el espacio latente entrenado por el algoritmo de clasificación, obteniendo 42 agrupaciones. Mediante evaluación de pertenencia (Fig. 6.7.a y Fig. 6.7.b) se identificaron 15 agrupaciones sobre las cuales tienden a agruparse algunas de las etiquetas originales (considerando la pertenencia de al menos el 20% de una etiqueta original dentro del cluster) y, utilizando esa condición, estas 15 se reducen a 9 agrupaciones (identificadas como “cXX” en Fig. 6.13 y Fig. 6.14), quedando las 27 restantes identificadas como “dXX” en Fig. 6.13 y Fig. 6.14), siendo un total de 36 nuevas etiquetas.

Se procedió a re-etiquetar el corpus, utilizando la clase asignada por expertos utilizando BIO-tagging para las entidades de más de una palabra, y utilizando la etiqueta obtenida mediante el procedimiento de clustering para las palabras sin etiqueta (que denominamos “Output” o “None”). Esta se trata de una tarea “falsa” para el reentrenamiento del modelo del estado del arte utilizando las

nuevas etiquetas, que buscaba reducir el desempeño sobre las palabras no-etiquetadas, que se encuentran fuera de las clases de interés, y que, finalmente, representaban la mayor parte del corpus.

Se evaluó el desempeño sobre las clases de interés originales del corpus (Fig. 6.15), Obteniendo como resultados: Pr: 80%, Re: 78% y F1-score: 79%, como promedio macro de la tarea, sin incluir la entidad "None", y: Pr: 81%, Re: 80% y F1-score: 81%, como promedio macro de la tarea, incluyendo a las palabras no etiquetadas. Evaluando el promedio macro inverso de la tarea, para las clases de interés, se obtiene: Pr: 77%, Re: 77% y F1-score: 77%

A partir de los resultados originales y al modelo re-etiquetado, se observa un aumento en el *Recall* promedio sobre las clases de interés de un 76% a un 78%. Es de interés el aumento en clases con menor presencia en el corpus como "*Duration*" con un *Recall* que aumenta del 70% al 81%, y "*ADE*", que aumenta del 32% al 38%. Se observa una disminución en la Precision promedio macro de la tarea (de un 83% a un 80%), esto implica un aumento en los falsos positivos, es decir, palabras erróneamente etiquetadas para cierta etiqueta, sin corresponder a esta. Sin embargo, este efecto tiene sentido, en el marco que la mayor confusión existente dentro de la que, anteriormente, era la clase más predominante. Bajo este nuevo escenario, es más fácil que las palabras que no eran de interés sean confundidas por una clase de interés (disminución de la *Precision*) que las palabras de interés sean confundidas como "None" (aumento del *Recall*). Considerando el promedio macro y macro-sin-None de la tarea completa, estos se mantienen en 81% y 79%, respectivamente. En los resultados generales de la tarea NER se observa una mejora porcentual del F1-score al considerar el promedio macro inverso de la tarea sobre las clases de interés, de 75% a 77%. A partir de lo anterior, es posible identificar que el modelo re-etiquetado tiende a presentar una mejora, principalmente para clases etiquetadas con menor presencia en el corpus, y que se ven afectadas por el desbalance de clases.

Capítulo 7. Discusión y Conclusiones

En el presente trabajo de investigación se analizó un corpus de 1089 documentos clínicos de texto libre previamente etiquetados por expertos, se realizó pre-procesamiento mediante su normalización, tokenización y representación de etiquetas mediante BIO-Tagging. Se replicó y entrenó el modelo propuesto por el estado-del-arte mediante el uso de Keras. Se re-etiquetó el corpus mediante un algoritmo de clústering de aprendizaje no supervisado, como una tarea "falsa" para el reentrenamiento del modelo del estado del arte utilizando las nuevas etiquetas. Se evaluó el desempeño sobre las clases originales del corpus:

En comparación al modelo original del estado del arte, se observó un incremento promedio del Recall sobre las clases de interés (etiquetas por expertos) de un 76% a un 78%, con una disminución sobre la Precisión (de un 83% a un 80%). Esto se puede explicar debido al aumento del número de clases en la clasificación, además de la subdivisión de la clase "*None*" que representaba cerca del 80% del texto, lo cual facilitó enfrentar el desbalance de clases, disminuyendo la confusión de las clases de interés a las clases de mayor presencia en el corpus.

Se observa un aumento en los casos de verdaderos positivos sobre las clases menos presentes dentro del corpus, como, por ejemplo, "*Duration*" con un "*Recall*" relación de 3:1000, frente al total de palabras sin etiquetar en el texto original. Por otra parte, una de las clases de mayor interés, correspondiente a la de eventos de efectos adversos por medicamentos ("*ADE*"), igualmente es la que presenta los resultados más bajos en la clasificación. Esto, debido a que presenta una mayor confusión con las clases "*SSLIF*" e "*IND*", debido a vocabulario compartido y diferenciación principalmente debido al contexto, además de ser una de las clases con menores ejemplos dentro del corpus, con 619 palabras dentro del conjunto de pruebas: Una relación de 1:2 frente a la etiqueta "*IND*", 6:100 frente a la etiqueta "*SSLIF*", y de 6:1000 frente al conjunto de palabras no etiquetadas.

Finalmente, se observa una mejora porcentual del F1-score promedio macro inverso de la tarea completa sobre las clases de interés, de 75% a 77%.

7.1. Trabajo Futuro

Con respecto a algunas de las ideas desechadas en las etapas iniciales de la investigación, como el uso de un autoencoder para codificar las secuencias de textos, las características entrenadas

demostraron ser informativas y permitir la clasificación. Sin embargo, presentaron con un desempeño menor al utilizar el *Word-embedding* directamente. Una posible mejora para esta aproximación podría ser aplicar un método que utilice una función de costo que le entregue más importancia al vocabulario del contexto en la reconstrucción, que a la palabra a etiquetar.

Algunas de las posibles mejoras directamente al modelo del estado del arte, podrían ser la inclusión de un *character-embedding* que permita incluir la información morfológica de los términos que no se encuentran en vocabulario de entrenamiento. Dentro de las propuestas para la arquitectura del modelo, considerando un modelo exclusivamente para los datos utilizados en esta investigación, se podría desarrollar de un modelo de etiquetado jerárquico, de acuerdo a las características de las etiquetas del corpus, separando *Entidades* de *No-Entidades*, posteriormente *SSD (efectos secundarios y síntomas)* de *Medication (medicamentos)*, y luego por la entidad individual correspondiente.

El corpus presentaba un alto desbalance de clases con cerca de un 80% del corpus correspondiente a palabras sin etiquetar. Mediante al uso de un algoritmo de clústering, se identificaron patrones dentro de este conjunto de palabras, que podrían representar nuevas clases a ser descubiertas o etiquetadas por expertos. Una posible extensión de la presente línea de investigación podría basarse en buscar un grupo de expertos para analizar y etiquetar las secuencias de palabras formadas por los clústeres descubiertos dentro de los documentos. Con el fin de generar un nuevo Gold-Estándar sobre el corpus, para un entrenamiento más específico de la tarea NER.

La tarea de reconocimiento de entidades en texto libre no es una tarea trivial, menos aún, trabajando con documentos de mayor extensión, con alta riqueza de vocabulario, pero al mismo tiempo, tan específicos como los utilizados en el historial clínico. Sin embargo, esta complejidad nos habla al mismo tiempo del desafío que supone, y de las oportunidades que conllevan para el uso de nuevas herramientas del aprendizaje automático e informática médica. El futuro de la salud estará definido por nuestra capacidad de utilizar estas herramientas, no solo para proponer nuevas y más eficientes soluciones, sino que, para tomar decisiones acertadas y correctas que puedan generar un impacto directo en la salud de las personas.

Referencias

[1] Reinsel D., Gantz J., Rydning J. (2018) "Data Age 2025: The Digitization of the World From Edge to Core" IDC White Paper, Seagate (November 2018) [Available online: Aug 2020] [Link: https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf]

[2] Transparency Market Research (TMR), (2018) "Electronic Health Records Market: Increasing Popularity of E-Prescription to Fuel Market Demand" Feb, 2018 [Available online: Aug 2020] [Link: https://www.transparencymarketresearch.com/pressrelease/electronic-health-records-market.htm]

[3] "World Health Organization International (2020) "COVID-19 and digital health: What can digital health offer for COVID-19?" [Available online: Aug 2020] [Link: https://www.who.int/china/news/feature-stories/detail/covid-19-and-digital-health-what-can-digital-health-offer-for-covid-19/]

[4] Plan de e-Salud (s.f.), Estrategia Digital – Objetivos estratégicos, Chile [Link: http://www.salud-e.cl/plan/objetivos-estrategicos/]

[5] Ministerio de Salud (2017) "Chiletec-Day: Salud avanza en la digitalización del sector" [Available online: Aug 2020] [Link: http://www.minsal.cl/chiletec-day-salud-avanza-en-la-digitalizacion-del-sector/]

[6] E Health Reporter (2018) Soledad Muñoz: "La estrategia digital en salud es un camino a largo plazo que trasciende a los gobiernos; es una visión de Estado" [Available online: Aug 2020] [Link: http://ehealthreporter.com/es/noticia/soledad-munoz-la-estrategia-digital-en-salud-es-un-camino-a-largo-plazo-que-trasciende-a-los-gobiernos-es-una-vision-de-estado/]

[7] D'Avolio L. (2017) "The Role of Machine Learning in Making EHRs Worth It" Article from "Towards Data Science", Jul 20, 2017 [Available online: Aug 2020] [Link: https://towardsdatascience.com/the-role-of-machine-learning-in-making-ehrs-worth-it-3d22ece8ede5]

[8] Official Website, The Office of the National Coordinator for Health Information Technology (s.-f.) "What is an electronic health record (EHR)?" [Available online: Aug 2020] [Link: https://www.healthit.gov/faq/what-electronic-health-record-ehr]

[9] Jagannatha, A. N., & Yu, H. (2016). Structured prediction models for RNN based sequence labeling in clinical text. Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing, 2016, 856–865. [Available online: Aug 2020] [Link: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5167535/]

[10] Jagannatha, A. N., & Yu, H. (2016). Bidirectional RNN for Medical Event Detection in Electronic Health Records. Proceedings of the Conference. Association for Computational Linguistics. North American Chapter. Meeting, 2016, pp. 473–482. [Available online: Aug 2020] [Link: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5119627/]

[11] T. Poibeau et al. (2013), "Multi-source, Multilingual Information Extraction and Summarization 11, Theory and Applications of Natural Language Processing", Springer-Verlag Berlin Heidelberg 2013, Chapter 2, Information Extraction: Past, Present and Future, pp. 23-44

[12] Shepherd G (2012) "Adverse drug reaction deaths reported in United States vital statistics, 1999-2006." Annals of Pharmacotherapy Vol 46, Issue 2, pp. 169 – 175, January 17, 2012 [Available online: Aug 2020] [Link: https://www.ncbi.nlm.nih.gov/pubmed/22253191]

[13] El Mostrador (2014) Jaime Mañalich, "Muertes en Melipilla: hospitales peligrosos" [Available online: Aug 2020] [Link: http://www.elmostrador.cl/noticias/opinion/2014/08/07/muertes-en-melipilla-hospitales-peligrosos/]

[14] Mena P. (2008) "Error médico y eventos adverso" Pontificia Universidad Católica de Chile, Revista Chilena de Pediatría 2008; 79 (3): 319-326 [Available online: Aug 2020] [Link: https://scielo.conicyt.cl/pdf/rcp/v79n3/art12.pdf]

[15] Loglisci C., et al. (2009) "A Knowledge-Based Framework for Information Extraction from Clinical Practice Guidelines", J. Rauch et al. (Eds.): ISMIS 2009, LNAI 5722, pp. 119–128, 2009.

[16] Settles B. (2004) "A Biomedical Named Entity Recognition Using Conditional Random Fields and Rich Feature Sets" Proceedings of the COLING 2004 International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA). Geneva, Switzerland. 2004. DOI:10.1.1.112.7693

[17] University of Massachusetts Lowell, Worcester, Amhers (2018) "MADE 1.0 – Challenge Official Website" [Available online: Aug 2020] [Link: https://bio-nlp.org/index.php/projects/39-nlp-challenges]

[18] Jurafsky D., Martin J. H., (2017) Chapter 21, "Information Extraction", pp. 348–375, Libro "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition"

Third Edition, Stanford University, University of Colorado at Boulder, August 28, 2017 [Available online: Aug 2020] [Link: https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf]

[19] Cohen K. (2014), "Methods in Biomedical Informatics", Academic Press, Oxford, 2014, Chapter 6 - Biomedical Natural Language Processing and Text Mining, Pages 141-177, ISBN 9780124016781, [Available online: Aug 2020] [Link: https://www.sciencedirect.com/science/book/9780124016781]

[20] Jurafsky D., Martin J. H., (2017) Chapter 1, "Regular Expressions, Text Normalization, Edit Distance" pp. 10–32, Libro "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition" Third Edition, Stanford University, University of Colorado at Boulder, August 28, 2017 [Available online: Aug 2020] [Link: https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf]

[21] Natural Language Toolkit (2017) NLTK 3.3 documentation [Available online: Aug 2020] [Link: http://www.nltk.org/]

[22] Jurafsky D., Martin J. H., (2017) Chapter 10, "Part-of-Speech Tagging" pp. 142–166. Libro "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition" Third Edition, Stanford University, University of Colorado at Boulder, August 28, 2017 [Available online: Aug 2020] [Link: https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf]

[23] Jurafsky D., Martin J. H., (2017) Chapter 6 "Evaluation: Precision, Recall, F-measure", "More than two classes" "Test sets and Cross-validation" pp. 83–90, Libro, "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition" Third Edition, Stanford University, University of Colorado at Boulder, August 28, 2017 [Available online: Aug 2020] [Link: https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf]

[24] Scikit-learn Documentation (s.f.) [Available online: Aug 2020] [Link: http://scikit-learn.org/stable/index.html]

[25] Chiticariu L. et al. (2013) "Rule-based Information Extraction is Dead! Long Live Rule-based Information Extraction Systems!", Association for Computational Linguistics, Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 827–832, Seattle, Washington, USA, 18-21 October 2013 [Available online: Aug 2020] [Link: http://www.aclweb.org/anthology/D13-1079]

[26] Tzong-Han et al. (2006) "NERBio: using selected word conjunctions, term normalization, and global patterns to improve biomedical named entity recognition", Volume 7, Supplement 5, APBioNet – Fifth International Conference on Bioinformatics (InCoB2006) [Available online: Aug 2020] [Link: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-7-S5-S11]

[27] Lee et al. (2004) "Biomedical named entity recognition using two-phase model based on SVMs", Journal of Biomedical Informatics, Volume 37, Issue 6, 2004, Pages 436-447, [Available online: Aug 2020] [Link: http://www.sciencedirect.com/science/article/pii/S1532046404000863]

[28] Kazama J. et al. (2002). "Tuning support vector machines for biomedical named entity recognition". In Proceedings of the ACL-02 workshop on Natural language processing in the biomedical domain - Volume 3 (BioMed '02), Vol. 3. Association for Computational Linguistics, Stroudsburg, PA, USA, 1-8. [Available online: Aug 2020] [Link: https://doi.org/10.3115/1118149.1118150]

[29] Ponomareva (2007) "Conditional Random Fields vs. Hidden Markov Models in a biomedical Named Entity Recognition task", Universidad Politecnica de Valencia, España [Available online: Aug 2020] [Link: http://clg.wlv.ac.uk/papers/Ponomareva-RANLP-07.pdf]

[30] Lafferty J. et al (2001) "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data" Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001), pages 282-289. [Available online: Aug 2020] [Link: https://repository.upenn.edu/cgi/viewcontent.cgi?article=1162&context=cis_papers]

[31] Zhou G. (2002) "Named Entity Recognition using an HMM-based Chunk Tagger" Laboratories for Information Technology, Singapore, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 473-480. [Available online: Aug 2020] [Link: http://www.aclweb.org/anthology/P02-1060]

[32] Zhao S. (2004) "Named Entity Recognition in Biomedical Texts using an HMM Model", Department of Computing Science University of Alberta Edmonton, Canada, [Available online: Aug 2020] [Link: http://www.aclweb.org/anthology/W04-1216]

[33] Settles B. (2004) "Biomedical Named Entity Recognition Using Conditional Random Fields and Rich Feature Sets" University of Wisconsin-Madison, Proceedings of the COLING 2004 International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA). Geneva, Switzerland. [Available online: Aug 2020] [Link: doi=10.1.1.112.7693]

[34] Huang (2015) "Bidirectional LSTM-CRF Models for Sequence Tagging", Computer Science - Computation and Language, eprint, Agosto 2015. [Available online: Aug 2020] [Link: https://arxiv.org/abs/1508.01991]

[35] Deng Li, Yu D. (2013) “Deep Learning Methods and Applications” - Foundations and Trends in Signal Processing Vol. 7, Nos. 3–4 (2013) 197–387. [Available online: Aug 2020] [Link: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/DeepLearning-NowPublishing-Vol7-SIG-039.pdf]
[36] Haykin S. (2009) “Neural Networks and Learning Machines”. Pearson Education, Upper Saddle River, NJ, 3rd edition, [Available online: Aug 2020] [Link: http://dai.fmph.uniba.sk/courses/NN/haykin.neural-networks.3ed.2009.pdf]
[37] Goodfellow I. et al. (2016) “Deep Learning”, MIT Press Book. Part II: Modern Practical Deep Networks. pp 166-478 [Available online: Aug 2020] [Link: http://www.deeplearningbook.org/]
[38] Hristev, R. M. (1998). “The ANN Book” [Available online: Aug 2020] [Link: http://neuron.tuke.sk/hudecm/PDF_PAPERS/Hritsev_The_ANN_Book.pdf]
[39] Elman, J. L. (1990). Finding structure in time. Cognitive Science, 14:179–211. [Available online: Aug 2020] [Link: https://crl.ucsd.edu/~elman/Papers/fsit.pdf]
[40] S. Hochreiter and J. Schmidhuber. (1997) Long short-term memory. Neural Computation, 9(8):1735–1780, 1997. [Available online: Aug 2020] [Link: http://www.bioinf.jku.at/publications/older/2604.pdf]
[41] DL4J, (2017) “A Beginner’s Guide to Recurrent Networks and LSTMs”, Skymind. [Available online: Aug 2020] [Link: https://deeplearning4j.org/lstm.html#a-beginners-guide-to-recurrent-networks-and-lstms]
[42] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." The Journal of Machine Learning Research 15.1 (2014): 1929-1958.
[43] Ioffe. S (2015) “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift” Computer Science - Machine Learning, ARXIV [Available online: Aug 2020] [Link: https://arxiv.org/abs/1502.03167]
[44] Berry M. (1995) “Computational Methods for Intelligent Information Access” Department of Computer Science, University of Tennessee. [Available online: Aug 2020] [Link: http://web.eecs.utk.edu/~mberry/sc95/sc95.html]
[45] Jiang Z., and Li, L., et al. (2015) “Training word embeddings for deep learning in biomedical text mining tasks” 2015 IEEE International Conference on Bioinformatics and Biomedicine (BTBM), [Available online: Aug 2020] [Link: http://doi.ieeecomputersociety.org/10.1109/BIBM.2015.7359756]
[46] Mikolov T., et al., (2013) “Distributed representations of words and phrases and their compositionality," in Advances in Neural Information Processing Systems, 2013, pp. 3111-3119. [Available online: Aug 2020] [Link: https://arxiv.org/pdf/1310.4546.pdf]
[47] Mikolov T., et al., (2013) “Efficient Estimation of Word Representations in Vector Space” [Available online: Aug 2020] [Link: https://arxiv.org/pdf/1301.3781.pdf]
[48] J. Pennington, R. Socher, and e. Manning, “Glove: Global vectors for word representation” Proc. Empiricial Methods Nat. Lang. Process. (EMNLP 2014),2014. [Available online: Aug 2020] [Link: https://www.aclweb.org/anthology/D14-1162]
[49] Mikolov T., et al (2013) “Linguistic Regularities in Continuous SpaceWord Representations” Proceedings of NAACL-HLT 2013, pages 746–751, Atlanta, Georgia, 9–14 June 2013. Association for Computational Linguistics [Available online: Aug 2020] [Link: https://www.aclweb.org/anthology/N13-1090]
[50] Meyer D., (2016) “How exactly does word2vec work?” [Available online: Aug 2020] [Link: http://www.1-4-5.net/~dmm/ml/how_does_word2vec_work.pdf]
[51] Bishop C. M (2006) Pattern Recognition and Machine Learning, Springer New York, 2016, 9.1 “9.1. K-means Clustering”, pp 424-429
[52] Zhao, Y., & Karypis, G. (2002). Comparison of agglomerative and partitional document clustering algorithms (No. TR-02-014). Minnesota Univ Minneapolis Dpt of Computer Science.
[53] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, *20*, 53-65.
[54] Kundeti S. (2016) “Clinical Named Entity Recognition: Challenges and opportunities” IEEE International Conference on Big Data (Big Data) 2016 [Available online: Aug 2020] [Link: doi:10.1109/BigData.2016.7840814]
[55] Yao L. (2015) “Biomedical Named Entity Recognition based on Deep Neutral Network” International Journal of Hybrid Information Technology – Vol.8, No.8 (2015), pp.279-288 [Available online: Aug 2020] [Link: http://dx.doi.org/10.14257/ijhit.2015.8.8.29]
[56] Lyu C., et al. (2017) “Long short-term memory RNN for biomedical named entity recognition” BMC Bioinformatics – Open Access [Available online: Aug 2020] [Link: DOI: 10.1186/s12859-017-1868-5]
[57] Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., & Lai, J. C. (1992). Class-based n-gram models of natural language. Computational linguistics, 18(4), 467-479.
[58] Tang B. (2014) “Evaluating Word Representation Features in Biomedical Named Entity Recognition Tasks” Hindawi Publishing Corporation, BioMed Research International, Volume 2014. [Available online: Aug 2020] [Link: http://dx.doi.org/10.1155/2014/240403]

[59] Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., & Lai, J. C. (1992). Class-based n-gram models of natural language. Computational linguistics, 18(4), 467-479.
[60] Habibi M. (2017) "Deep learning with word embeddings improves biomedical named entity recognition" ISMB/ECCB 2017, Bioinformatics, 33, 2017, i37–i48 [Available online: Aug 2020] [Link: doi: 10.1093/bioinformatics/btx228]
[61] Jagannatha, A., Liu, F., Liu, W., & Yu, H. (2019). Overview of the First Natural Language Processing Challenge for Extracting Medication, Indication, and Adverse Drug Events from Electronic Health Record Notes (MADE 1.0). Drug Safety. doi:10.1007/s40264-018-0762-z
[62] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. [Available online: Aug 2020] [Link: https://arxiv.org/abs/1810.04805]
[63] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI Blog, 1(8), 9. [Available online: Aug 2020] [Link: https://www.semanticscholar.org/paper/Language-Models-are-Unsupervised-Multitask-Learners-Radford-W]
[64] Yan, H., Deng, B., Li, X., & Qiu, X. (2019). Tener: Adapting transformer encoder for name entity recognition. arXiv preprint arXiv:1911.04474. [Available online: Aug 2020] [Link: https://arxiv.org/abs/1911.04474]
[65] Khan, M. R., Ziyadi, M., & AbdelHady, M. (2020). MT-BioNER: Multi-task Learning for Biomedical Named Entity Recognition using Deep Bidirectional Transformers. arXiv preprint arXiv:2001.08904. [Available online: Aug 2020] [Link: https://arxiv.org/abs/2001.08904]
[66] Rajapakse T. (2020) Simple Transformers Git-hub repository. [Available online: Aug 2020].[Link: https://github.com/ThilinaRajapakse/simpletransformers]
[67] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Agarwal, S. (2020). Language models are few-shot learners. arXiv preprint arXiv:2005.14165. [Available online: Aug 2020] [Link: https://arxiv.org/abs/2005.14165]
[68] Definición: "ABVD". (s.f) Instituto Nacional del cancer España [Available online: Aug 2020] [Link: https://www.cancer.gov/espanol/publicaciones/diccionario/def/abvd]
[69] Keras (s.f.) API Documentation [Available online: Aug 2020] [Link: https://keras.io/]
[70] TensorFlow (s.f.) Documentation [Available online: Aug 2020] [Link: https://www.tensorflow.org/api_docs/]
[71] Scikit-learn (s.f.) TSNE – Implentation [Available online: Aug 2020] [Link: https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html

Anexo A. IRB

Universidad de Concepción

Concepción, October 23rd, 2017.

CERTIFICATE

The Ethics Committee of the Universidad de Concepción has reviewed the proposal of the project granted in the ENLACE PROJECT CALL OF THE VICE-RECTORATE OF RESEARCH AND DEVELOPMENT OF THE UNIVERSIDAD DE CONCEPCION - Nº217.092.052-1.0, entitled "INTELLIGENT ALGORITHMS TO ASSIST IN CLINICAL DECISIONS AND EFFECTIVE DELIVERY OF HEALTH CARE INFORMATION, whose Responsible Investigator is the DRA. ROSA FIGUEROA ITURRIETA, academic associated to the Department of Electrical Engineering belong to the Faculty of Engineering of the University and has verified that it complies with the ethics and bioethics standards and procedures nationally and internationally established for studies in the area of health technologies, considering in this particular case, use of data from electronic clinical records.

In this scientific research proposal, we planned as main goal, to develop novel classification models based on automatically generated regular expressions algorithms that combine NLP, information extraction (IE) and a novel active learning decision criteria for selecting the most informative training samples to effectively extract key patient information and measurements embedded in clinical notes in Spanish

To initiate this research, it is necessary to obtain access to a set of data, previously de-identified, which consists of 1092 electronic clinical records annotated with the use of medications, indications, adverse events and other clinical notes from the UMass Memorial Health Care Inc. This set of data that will be released by The Massachusetts University, aims to conduct a public competition that seeks to develop natural language processing techniques (NLP, for its acronym in English) to detect drugs and adverse events in the electronic clinical files (NLP Challenges for Detecting Medication and Adverse Drug Events from Electronic Health Records (MADE1.0) via UMass BioNLP).

The data handling and extraction procedures and their subsequent analysis are rigorously and thoroughly described in the "Project Formulation", maintaining the due protection and confidentiality of the data that give rise to this proposal.

UNIVERSIDAD DE CONCEPCION
COMITÉ DE ÉTICA

Universidad de Concepción

The project presented observes the rights enshrined in the Universal Declaration of Human Rights, the rights and principles of the Universal Declaration on Bioethics and Human Rights, the Ethical Standards of the Pan American Health Organization for Research on Human Subjects. Likewise, it complies with the ethical and bioethical principles that should prevail in scientific research with the use of data from clinical records (de-identified data) in the area of studies related to human health and the application of new technologies, and with those established in the Singapore Declaration for Research Integrity, as established by the National Council for Scientific and Technological Research - CONICYT through Exempt Resolution Nº. 157, of January 24, 2013.

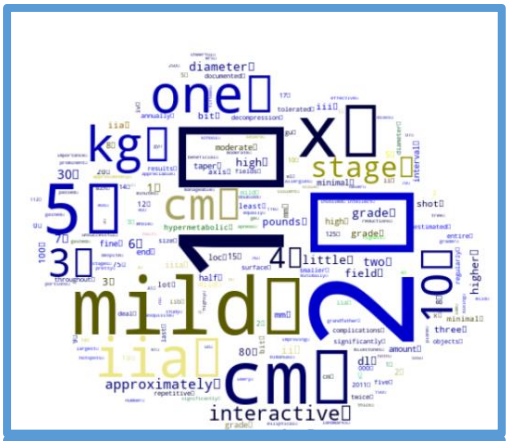
In view of the above and given that the proposed project does not show elements that may violate the norms and the guiding ethical principles of our University Institution and those adopted by the National Council of Scientific and Technological Research, this Committee resolves to approve it, conferring the present Certificate.

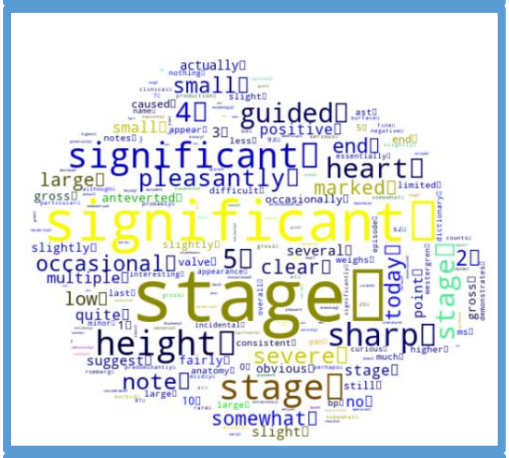
UNIVERSIDAD DE CONCEPCION
COMITÉ DE ÉTICA

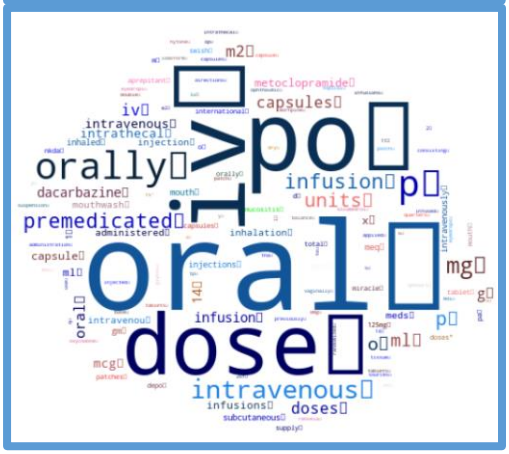
DR. JOSÉ BECERRA ALLENDE
CHAIR
ETHICS COMMITTEE
UNIVERSIDAD DE CONCEPCIÓN

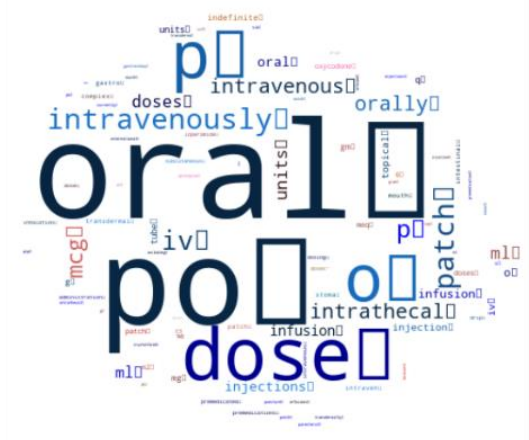
Anexo B. Nubes de Palabras

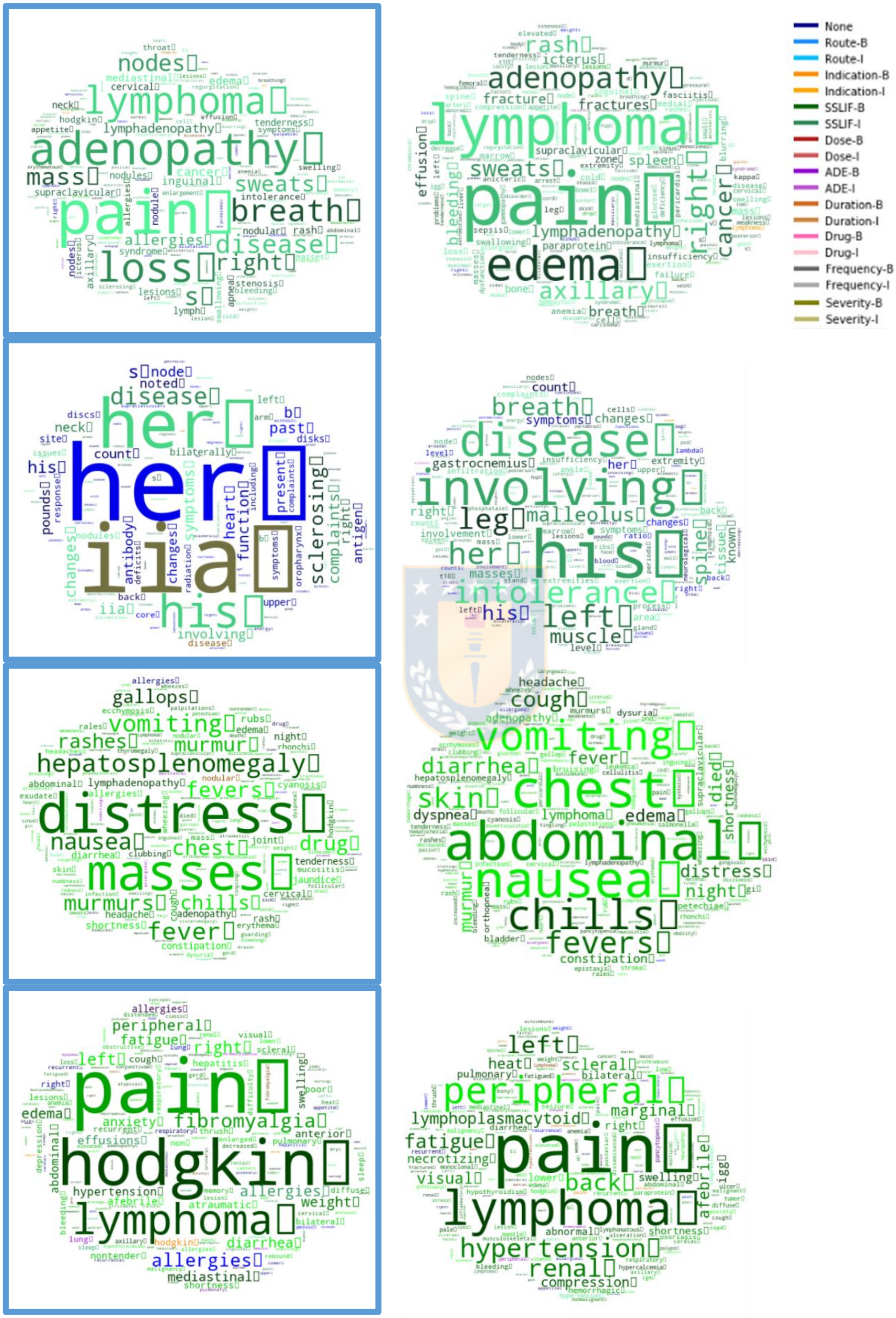
En la columna de la izquierda se presenta la nube de palabras formada para el clúster sobre el conjunto de entrenamiento y en la columna de la derecha la nube de palabras para el mismo clúster sobre el conjunto de validación. Se incluye una referencia de color para las diferentes etiquetas (de acuerdo a la etiqueta de expertos, duplicada por BIO-tagging). Se ordena desde las agrupaciones que presentan mayor pertenencia de parte de las clases de interés.

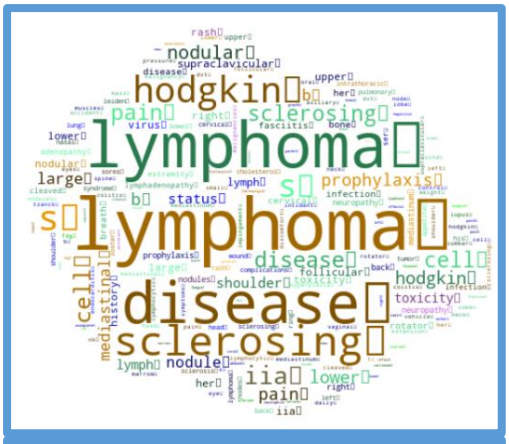




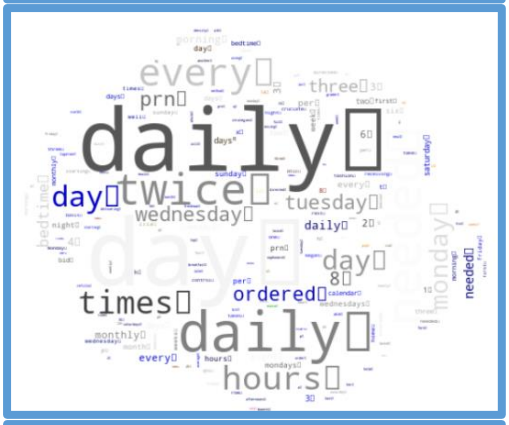


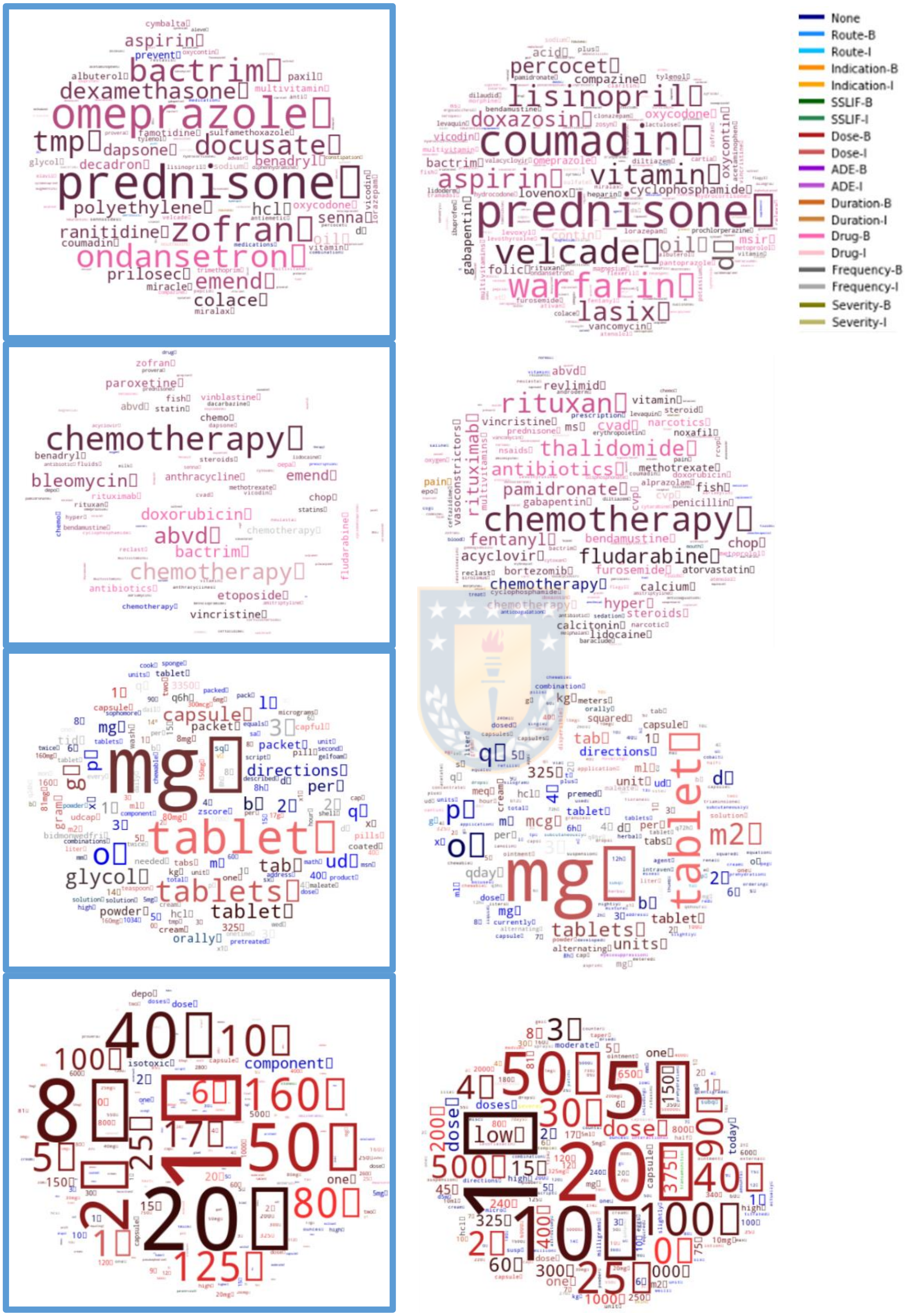


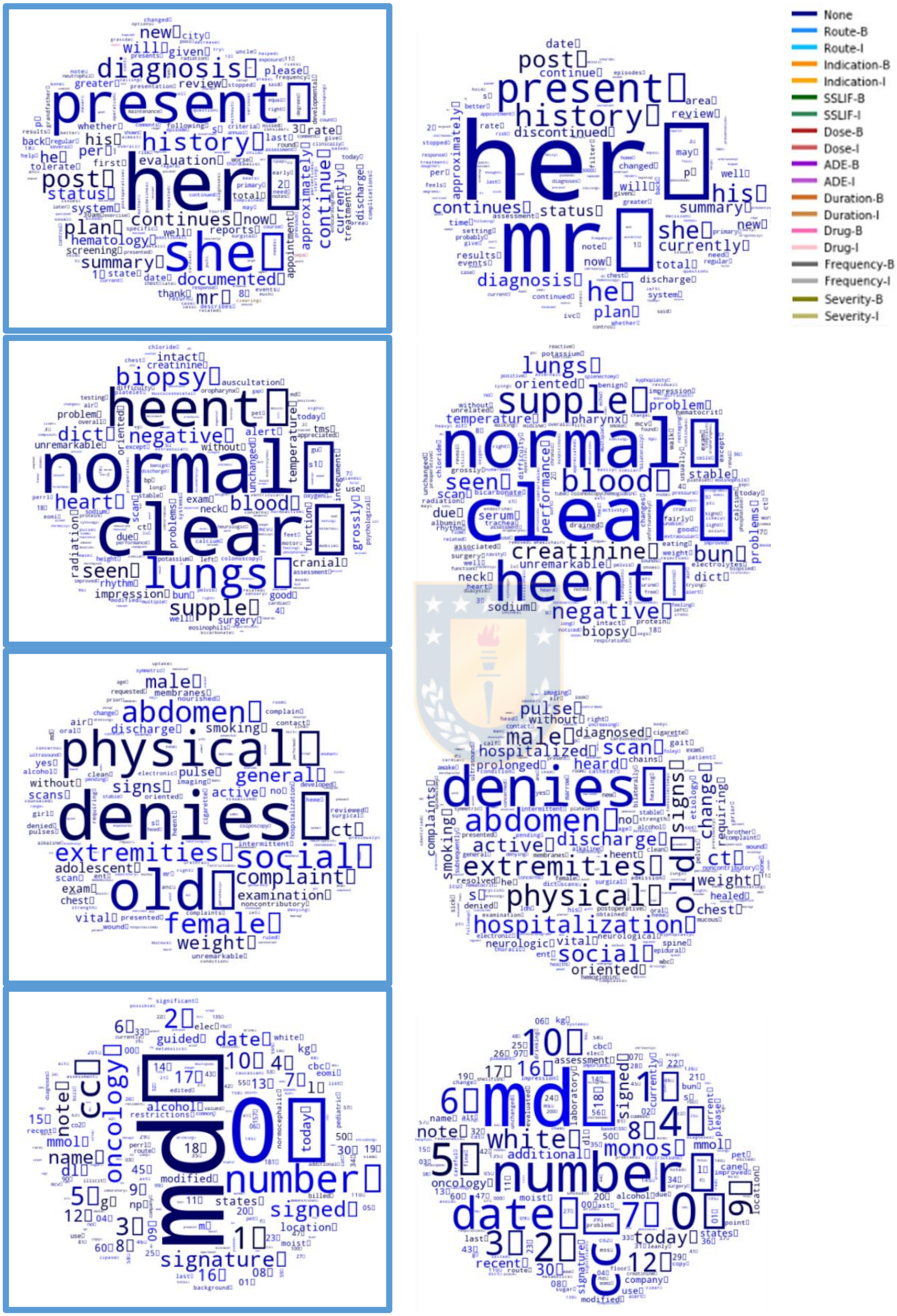

None
Route-B
Route-I
Indication-B
Indication-I
SSLIF-B
SSLIF-I
Dose-B
Dose-I
ADE-B
ADE-I
Duration-B
Duration-I
Drug-B
Drug-I
Frequency-B
Frequency-I
Severity-B
Severity-I

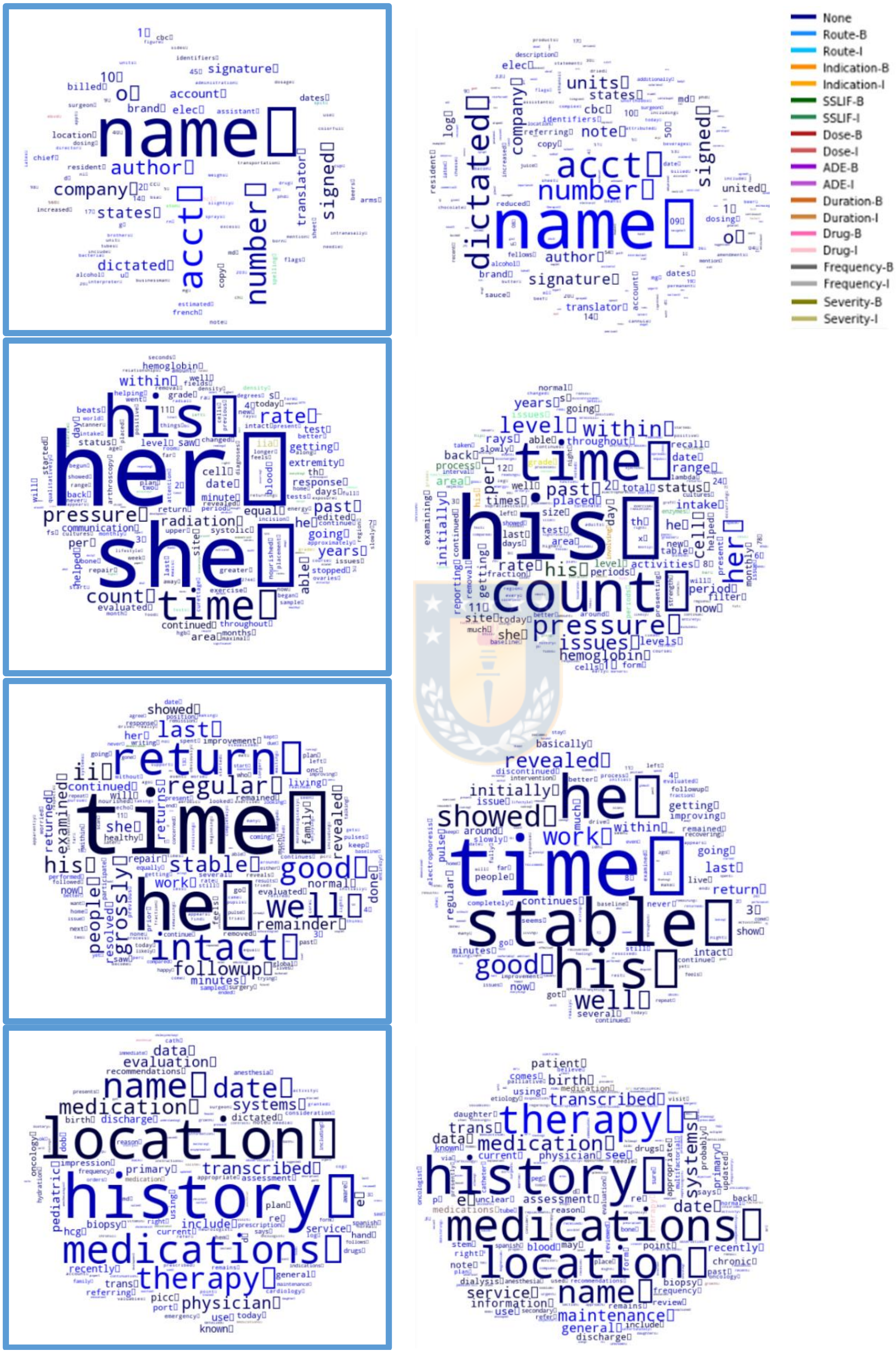


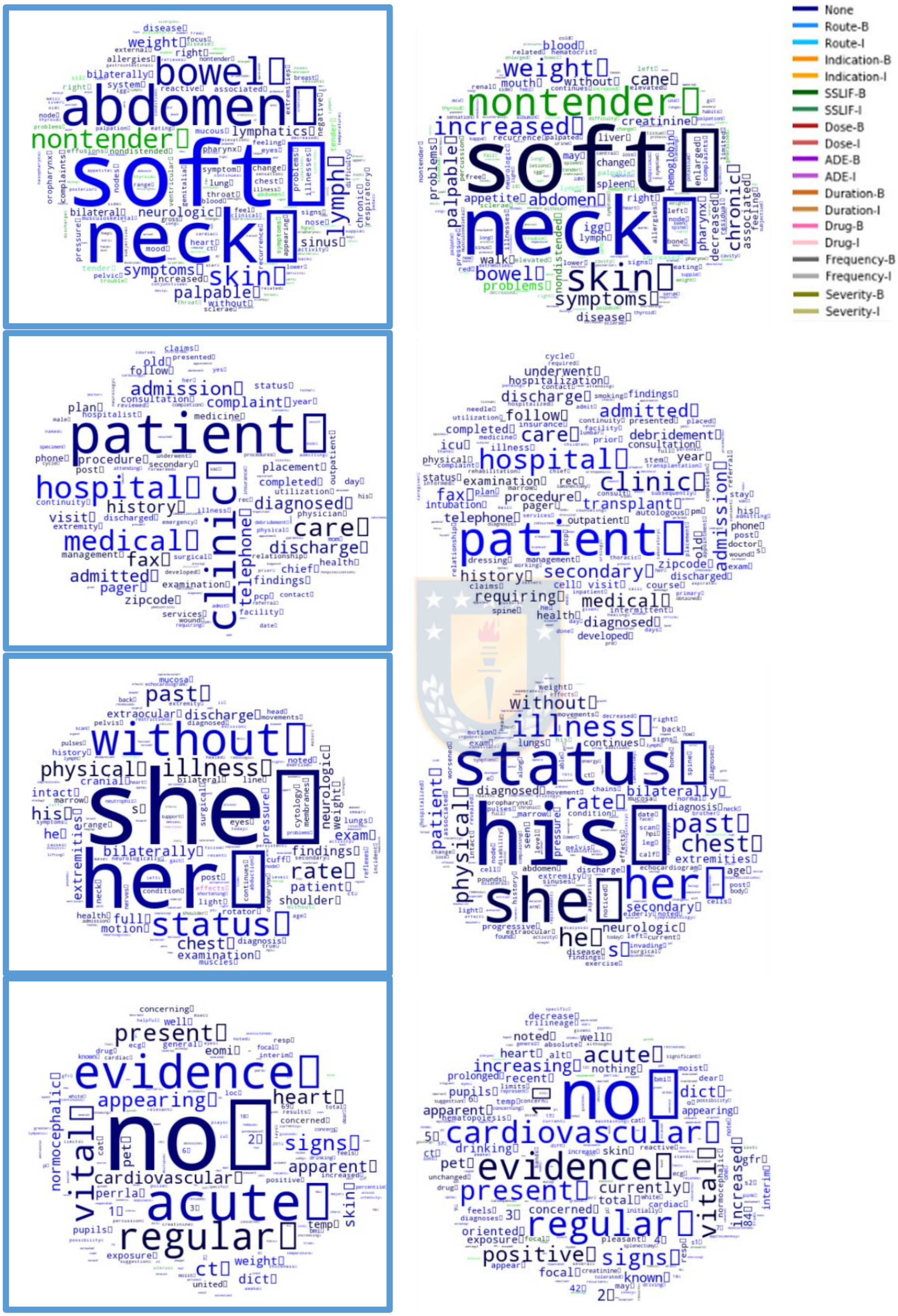
None
Route-B
Route-I
Indication-B
Indication-I
SSLIF-B
SSLIF-I
Dose-B
Dose-I
ADE-B
ADE-I
Duration-B
Duration-I
Drug-B
Drug-I
Frequency-B
Frequency-I
Severity-B
Severity-I




None
Route-B
Route-I
Indication-B
Indication-I
SSLIF-B
SSLIF-I
Dose-B
Dose-I
ADE-B
ADE-I
Duration-B
Duration-I
Drug-B
Drug-I
Frequency-B
Frequency-I
Severity-B
Severity-I

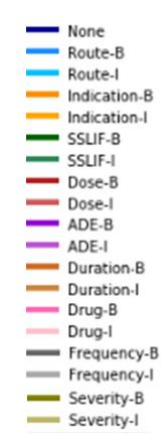

None
Route-B
Route-I
Indication-B
Indication-I
SSLIF-B
SSLIF-I
Dose-B
Dose-I
ADE-B
ADE-I
Duration-B
Duration-I
Drug-B
Drug-I
Frequency-B
Frequency-I
Severity-B
Severity-I


None
Route-B
Route-I
Indication-B
Indication-I
SSLIF-B
SSLIF-I
Dose-B
Dose-I
ADE-B
ADE-I
Duration-B
Duration-I
Drug-B
Drug-I
Frequency-B
Frequency-I
Severity-B
Severity-I


None
Route-B
Route-I
Indication-B
Indication-I
SSLIF-B
SSLIF-I
Dose-B
Dose-I
ADE-B
ADE-I
Duration-B
Duration-I
Drug-B
Drug-I
Frequency-B
Frequency-I
Severity-B
Severity-I

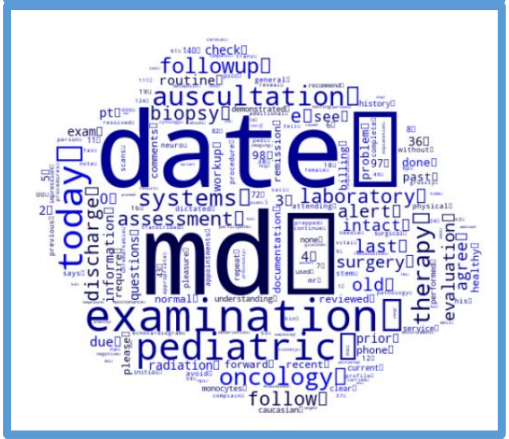

he
visit
cycle
pager
history
review
plan
fax
medical
post
home
will
social
primary
telephone
reviewed
mother
care
practitioner
help
receive
need
who
family
attending
assistant
protocol
diagnoses
instructions
date
received
continue
treated
following
caregiver
status
secondary
her
his
completed
discussed
supported
scheduled
call
taking
provider
case
discharged
tomorrow

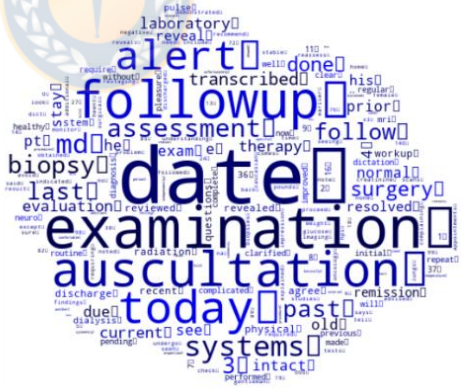

will
he
visit
plan
history
her
review
pager
post
help
fax
cycle
home
who
reviewed
need
she
his
patient
nurse
interim
provider
protocol
care
telephone
date
city
primary
following
call
received
continue
transfer
discharged
dr
daughter
family
past
male
attending
scheduled
secondary
status
caregiver
relief
ago
given
take
obtain
doctor
now
done
left

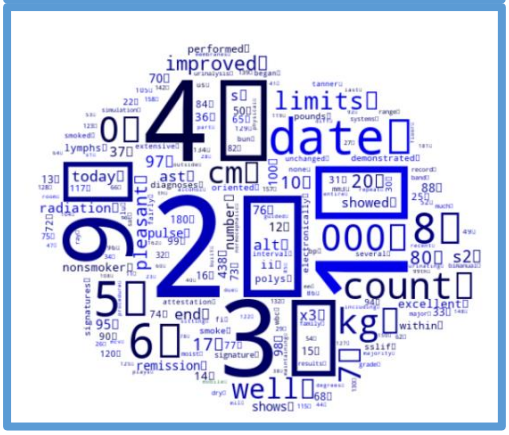

None
Route-B
Route-I
Indication-B
Indication-I
SSLIF-B
SSLIF-I
Dose-B
Dose-I
ADE-B
ADE-I
Duration-B
Duration-I
Drug-B
Drug-I
Frequency-B
Frequency-I
Severity-B
Severity-I


evaluated
mr
date
review
reviewed
p
b
reports
see
last
know
home
will
plan
time
interim
absolute
discussed
systems
transcribed
instructions
continue
current
previous
tolerated
given
female
regarding
documented
number
examined
regular
ongoing
recently
call
port
well
3
no

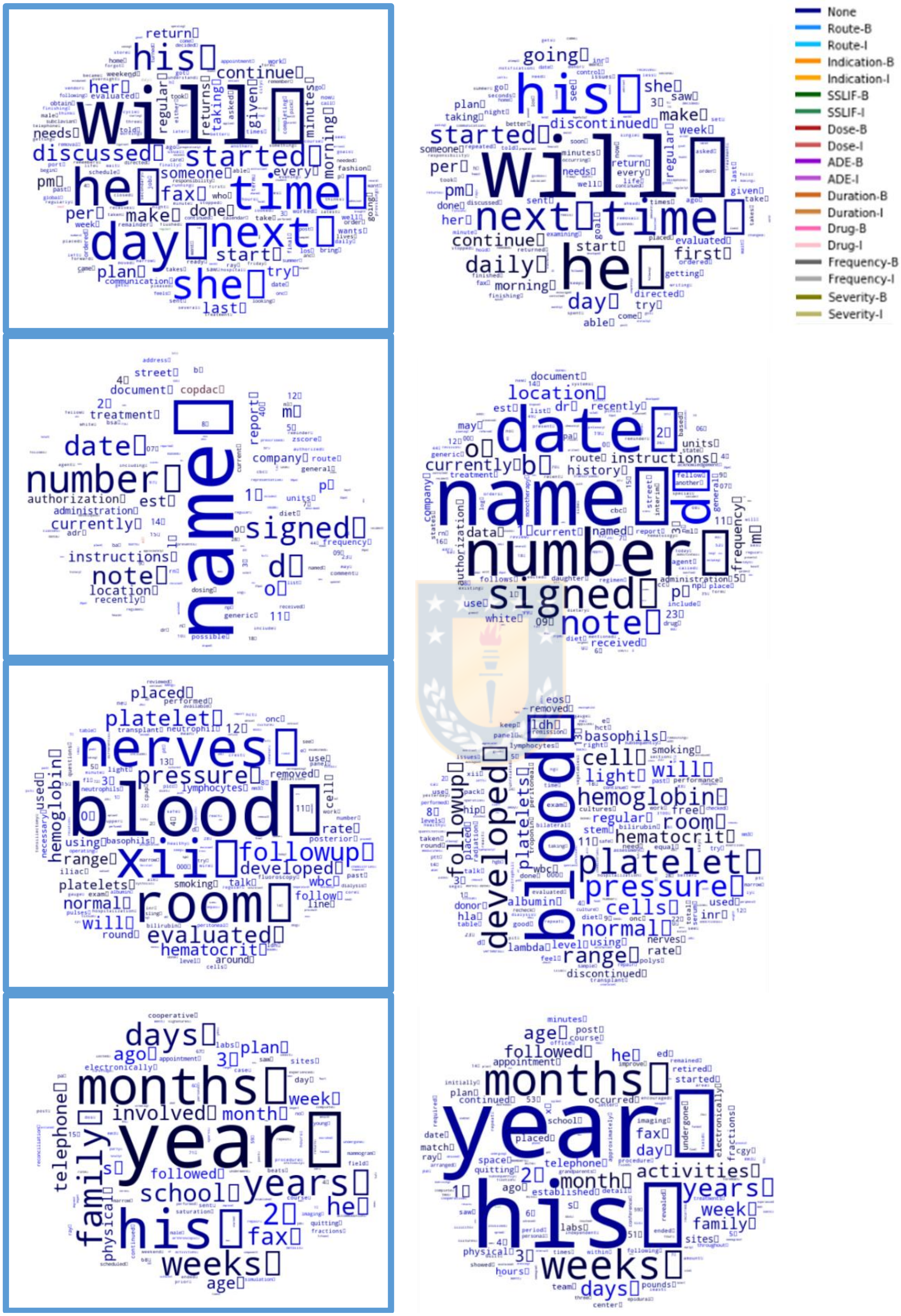

will
mr
evaluated
time
reports
see
home
p
date
now
point
no
diet
present
discussed
transcribed
current
well
plan
given
first
last
know
tolerated
review
interim
regular
systems
documented
family
consider
next
absolute
reviewed
instructions
ago
3
several

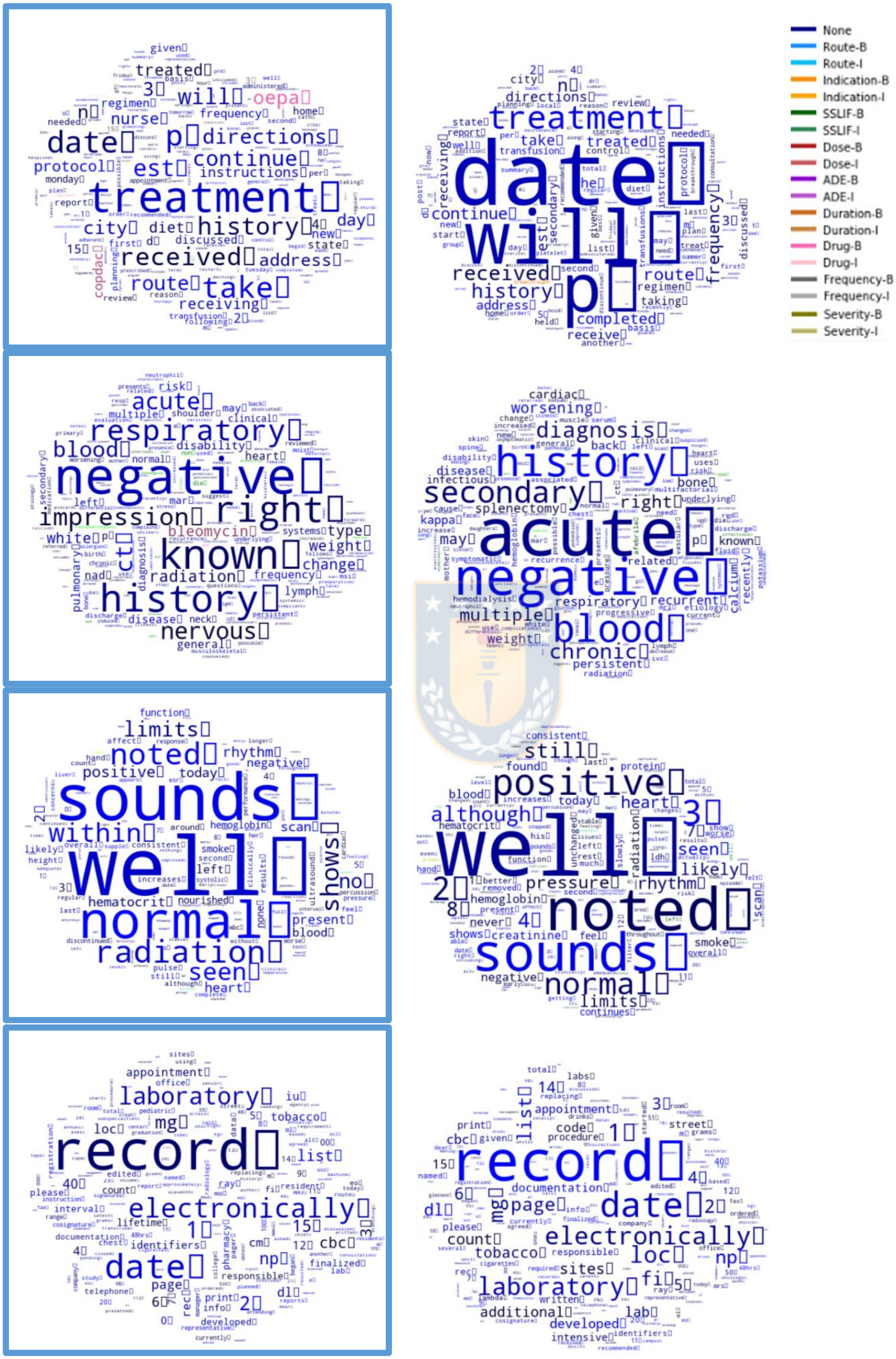

date
md
examination
pediatric
oncology
auscultation
today
discharge
systems
assessment
laboratory
therapy
biopsy
follow
followup
routine
check
alert
intact
last
surgery
old
prior
phone
radiation
forward
recent
normal
understanding
reviewed
done
past
problem
evaluation
agree
healthy
due
pleased
monocytes
complaint
caucasian

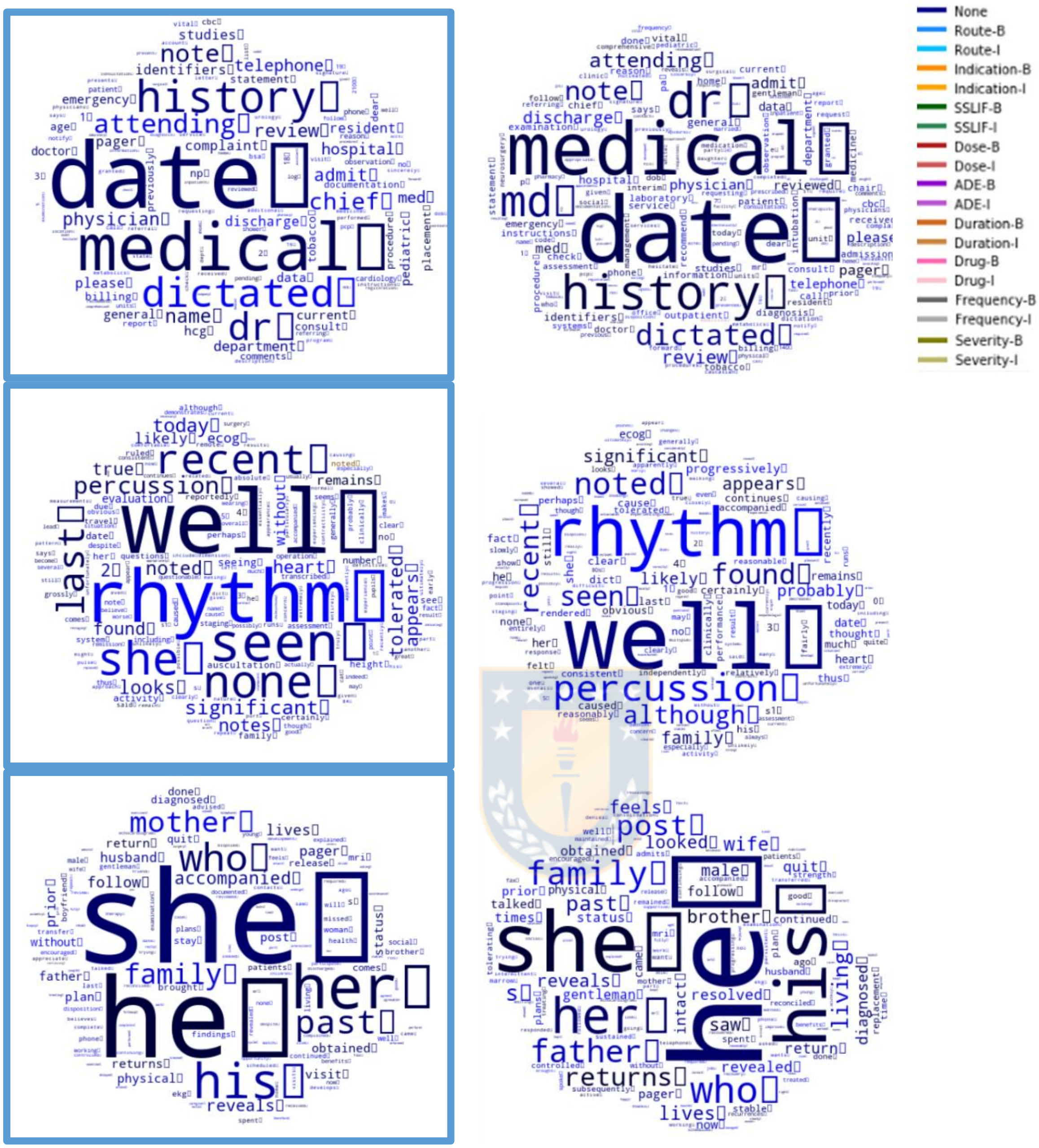

date
examination
auscultation
followup
alert
today
assessment
laboratory
systems
biopsy
md
done
transcribed
prior
follow
therapy
normal
surgery
resolved
remission
past
old
current
see
last
evaluation
reviewed
revealed
discharge
due
dialysis
pending
recent
complicated
physical
intact
3
exam
he
pt
workup
dictation


4
2
9
1
3
5
6
0
7
8
date
count
kg
well
limits
cm
improved
performed
pleasant
radiation
today
ast
pulse
nonsmoker
remission
end
shows
showed
excellent
within
smoked
s2
000
x3


2
date
count
1
3
9
0
4
5
6
improved
diagnoses
lymphs
pulse
polys
kyphoplasties
000
7
8
showed
ast
s2
ecg
noted
today
drive
one
numerous
pounds
limits
revealed
enzymes
blood
fairly
improving
unchanged
demonstrated
shows
smoked
last
bun
saturation
nonfasting
several
excellent
x3
20
kg
remission


None
Route-B
Route-I
Indication-B
Indication-I
SSLIF-B
SSLIF-I
Dose-B
Dose-I
ADE-B
ADE-I
Duration-B
Duration-I
Drug-B
Drug-I
Frequency-B
Frequency-I
Severity-B
Severity-I


None
Route-B
Route-I
Indication-B
Indication-I
SSLIF-B
SSLIF-I
Dose-B
Dose-I
ADE-B
ADE-I
Duration-B
Duration-I
Drug-B
Drug-I
Frequency-B
Frequency-I
Severity-B
Severity-I


None
Route-B
Route-I
Indication-B
Indication-I
SSLIF-B
SSLIF-I
Dose-B
Dose-I
ADE-B
ADE-I
Duration-B
Duration-I
Drug-B
Drug-I
Frequency-B
Frequency-I
Severity-B
Severity-I