UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

NESTED NAMED ENTITY RECOGNITION IN DIAGNOSES FROM THE CHILEAN WAITING LIST IN PUBLIC HOSPITALS

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS, MENCIÓN COMPUTACIÓN

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

MATÍAS IGNACIO ROJAS GALLEGOS

PROFESOR GUÍA:
FELIPE JOSÉ BRAVO MÁRQUEZ

PROFESOR CO-GUÍA:
JOCELYN MARIEL DUNSTAN ESCUDERO

MIEMBROS DE LA COMISIÓN:
BENJAMÍN EUGENIO BUSTOS CÁRDENAS
MARCELO GABRIEL MENDOZA ROCHA
MAURICIO DAVID CERDA VILLABLANCA

SANTIAGO DE CHILE
2022

Reconocimiento de entidades anidadas en diagnósticos de la lista de espera en hospitales públicos

En el sistema de salud público Chileno, las interconsultas realizadas por el médico general se presentan en forma de texto libre. Dentro de estos textos, podemos encontrar palabras (entidades) con relevancia clínica, como enfermedades, medicamentos, hallazgos clínicos, entre otros. La naturaleza no estructurada de estos textos, hace que el análisis manual sea complejo, incluso para los especialistas. Es por esto, que el desarrollo de un sistema de extracción automática de estas entidades, sería un importante apoyo tanto para la gestión de la lista de espera Chilena, así como el uso secundario de la información.

Con el propósito de desarrollar estos modelos, nuestro grupo de investigación utilizó el conocimiento experto para anotar entidades con relevancia clínica dentro de estos diagnósticos, consolidando así el corpus de la Lista de Espera Chilena. Este conjunto de datos contiene un alto porcentaje de entidades anidadas (46.7%), lo que constituye una tarea más conocida como el Reconocimiento de Entidades Nombradas Anidadas (NER anidado).

En esta tesis, utilizamos los avances recientes en aprendizaje profundo para desarrollar el modelo Multiple LSTM-CRF (MLC), un método capaz de reconocer entidades anidadas en nuestro corpus. Para validar su efectividad, llevamos a cabo un estudio empírico comparando nuestra arquitectura con varios modelos del estado del arte y otros datasets, prestando especial atención al impacto del uso de modelos del lenguaje pre-entrenados. Los resultados experimentales confirman la eficacia del modelo MLC, alcanzando el estado del arte en nuestro corpus con un micro F1-score de 80.5 y un rendimiento competitivo en el resto.

Adicionalmente, se proponen nuevas métricas de evaluación que nos permiten medir la capacidad de los modelos para detectar entidades anidadas, lo cual no ha sido abordado en trabajos previos. Los resultados señalan que la métrica de NER anidado no mide correctamente la capacidad de un modelo para detectar entidades anidadas, mientras que nuestras métricas proporcionan nuevas pruebas sobre cómo los enfoques existentes manejan la tarea. Finalmente, nuestro modelo fue incorporado a una página web, permitiendo que profesionales de la salud puedan probarlo y entregar retroalimentación para mejorar su rendimiento. Este trabajo constituye el primer intento de resolver la tarea de NER anidado en un corpus en Español, siendo además una herramienta importante para el estudio de la lista de espera.

Nested Named Entity Recognition in diagnoses from the Chilean Waiting List in public hospitals

In the public health system in Chile, general practitioner referrals are presented in the form of free text. Within these texts, we can find words (entities) with some clinical relevance, such as diseases, medications, clinical findings, among others. The unstructured nature of these texts makes manual analysis complex, even for specialists. Therefore, the development of an automatic extraction system of these entities could be an important support for both the management of the Chilean Waiting List and the secondary use of the information.

In order to develop these models, our research group used expert knowledge to annotate clinically relevant entities within these diagnoses, thus consolidating the Chilean Waiting List corpus. This dataset contains a high percentage of nested entities (46.7%), which constitutes a task known as Nested Named Entity Recognition (Nested NER).

In this thesis, we used recent advances in deep learning to develop the Multiple LSTM-CRF (MLC) model, a method capable of recognizing nested entities in our corpus. To validate its effectiveness, we conducted an empirical study comparing our architecture with several state-of-the-art models and other nested NER datasets, paying particular attention to the impact of using pre-trained language models. Experimental results confirm the effectiveness of the MLC model, achieving state-of-the-art in our corpus with a micro F1-score of 80.5 and competitive performance in the rest.

In addition, we proposed new evaluation metrics that allow us to adequately measure the model's ability to detect nested entities, which has not been addressed in previous work. The results indicate that the nested NER metric does not correctly measure the model's ability to detect nested entities, while our metrics provide new evidence on how existing approaches handle the task. Finally, our model was incorporated into a web page, allowing healthcare professionals to test it and provide feedback to improve its performance. This work constitutes the first attempt to solve the nested NER task in a Spanish corpus, being also an important support for the study of the Chilean Waiting List.

Dedicado a mi familia, amigos, Fran y su familia. Todos con su apoyo incondicional y cariño me han permitido llegar hasta acá.

Agradecimientos

Primero que nada, le agradezco con todo mi corazón a mis abuelos y a mi madre Verónica, quienes me criaron, educaron y me ayudaron a ser la persona que soy hoy en día, los amo mucho. Agradezco al amor de mi vida, Francisca, quién fue la persona más importante en este paso por la universidad. Gracias por todo el cariño y apoyo incondicional que ella y su familia me han entregado a lo largo de estos últimos 8 años.

En segundo lugar, me gustaría agradecer a mis profesores guías, Felipe y Jocelyn, por todo el apoyo que me han entregado en esta aventura en el mundo de la investigación. Sus conocimientos, experiencia y apoyo personal han facilitado mi paso por este Magister, además de permitirme aprender cosas nuevas cada día. Agradezco también al grupo de investigación PLN CMM, por entregarme la experiencia de trabajar en un grupo interdisciplinario y con grandes personas.

Le agradezco a Fernando y Vicente, mis amistades más cercanas durante mi vida, me han demostrado que las verdaderas amistades perduran en el tiempo independiente de la distancia. Agradezco de igual manera a los grandes amigos que formé en la universidad como Rodrigo, Mauricio y Emilio.

Doy las gracias también a los integrantes de mi comisión, los profesores Benjamín Bustos, Mauricio Cerda y Marcelo Mendoza, por todos sus comentarios y observaciones propuestas para mejorar esta tesis. Agradezco a Ren Cerro por sus correcciones en los artículos en Ingles. A Sandra y Angélica por su constante ayuda con mis múltiples dudas sobre el Magister.

Finalmente, agradezco el financiamiento recibido del Centro de Modelamiento Matemático (CMM), AFB170001, ACE210010 y FB210005 y los proyectos Fondecyt 11201250 y 11200290. Además, esta tesis fue parcialmente apoyada por la infraestructura de supercómputo del NLHPC (ECM-02).

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The Chilean health system is composed of a mixed public-private system that includes public insurance through the National Health Fund (FONASA) and insurance provided by a private institution (known as ISAPRE), where the former contains 77% of the Chilean population [9]. Primary health care represents the first contact level of individuals, families, and communities with the public health system, providing ambulatory assistance.

Statistics show that beneficiaries of FONASA present a high demand for visits to specialists, which is previously evaluated by general physicians in primary health care [82]. This demand problem is currently handled through the so-called Waiting List (WL). The WL is divided into "GES" (Spanish acronym for Explicit Health Guarantees) that covers 85 prioritized health conditions, and the "non-GES", which covers the remaining consultations. Nevertheless, the problem with this system lies in the long waiting times, which has severe consequences for the Chilean population. According to the information obtained through transparency law, about 15,665 patients died in 2020-01 while waiting for their first consultation with a specialist. In 2021, there were 1,965,653 people in the non-GES WL pending for a specialist's appointment, with an average waiting time above 501 days [31], which has increased to 543 days in 2021.

Every public health institution in Chile uploads weekly spreadsheets with information on GES and non-GES referrals. These referrals from general physicians contain data such as the patient's personal information, the healthcare provider that emits and receives the patient, the medical specialty, and the suspected diagnosis in the form of unstructured text. There are different sequences of words (entities) within these diagnoses, with medical relevance, such as diseases, laboratory results, therapeutic procedures, among others. The analysis of these entities could be used for epidemiological studies and the secondary use of the information. For example, it can support the prioritization of patients, the selection of cases that can be solved by telemedicine, the estimated number of people who present more than one disease (comorbidity) or that take more than one medication (polypharmacy), study the genetic burden of diseases, statistics of the pending procedures, or the family background of diseases when mentioned.

However, the manual extraction of these entities could be time-consuming, resource-intensive, and error-prone, even with skilled personnel. The main reasons are the extensive

use of non-standardized abbreviations, the variability of the clinical language across medical specialties and health professionals, and its restricted availability for privacy reasons, to mention some [25]. These difficulties can be efficiently addressed by implementing computational solutions to extract key information automatically.

Natural Language Processing (NLP) is a branch of artificial intelligence that deals with the interaction between humans and machines through language. The aim is to develop computational systems used for solving practical problems involving human language, better known as *NLP tasks*. In medicine, typical applications of NLP are text classification, detection of drug interactions, clinical concept extraction, automatic codification of diseases, or anonymization of electronic health records [25].

In our context, the task that better suits our problem is called Named Entity Recognition (NER), which aims to automatically identify essential pieces of information (named entities) in a text written in natural language. Most of the proposed methods for solving NER are based on neural networks, as they have recently demonstrated high performance in many NLP tasks. In particular, NER is commonly regarded as a sequence labeling problem, which assumes that each word has at most one associated label. This approach is known as Flat NER.

In order to train NER models, our research group collected non-GES referrals from 23 out of the 29 Chilean health services through the Transparency Law [87]. Then, using a specific annotation guide and the BRAT annotation software, the team have been manually annotated medical entities within these diagnoses using expert knowledge. To date, 5,000 diagnoses have been analyzed, thus consolidating the Chilean Waiting List corpus [11].

Compared to other corpora, this corpus has some characteristics that make it more challenging for the NER task. First, due to a lack of data and human resources, there are few studies on applying Named Entity Recognition models to Spanish clinical resources, such as ours. Second, this corpus has a high percentage of nested entities, which are entities contained within other entity mentions [33]. An example is "cancer de colon", where a Body Part (colon) is contained in a Disease. This task is better known as nested NER, and although several methods have been proposed to address the nesting problem, we realized that most of them rely on complex task-specific, ignoring some more intuitive and potentially useful baselines when comparing their approaches.

To address these issues, we follow two main lines of research in this work. First, we develop two simple, overlooked, yet powerful architectures to recognize nested entities in our medical corpus. Then, to validate the effectiveness of these methods, we conduct an empirical study comparing our models with several state-of-the-art nested NER architectures. These experiments are conducted in other corpora from different languages, focusing on the impact of using pre-trained language models. In the following section, we describe in detail the technical problem and challenges of automatically recognizing nested entities in our corpus.

1.1 Problem Statement

Recognizing entities in the Chilean Waiting List has significant challenges in terms of the nested NER task. First, the percentage of nested entities in this corpus (46.7%) is much more significant than other related corpora. Second, we found that most of the previous research in this task ignores the case in which the same span of text is tagged with more than one entity type, as shown in Figure 1.1. This case is very common in our corpus, and it was first noticed by Alex et al. [5] but was not analyzed further in the literature. Third, the corpus size is considerably smaller than other related corpora, which could affect the performance of deep learning models. In addition, to the best of our knowledge, no one has studied the nested NER task applied to Spanish resources. Hence, studying this corpus represents an opportunity to extend research on this task to other languages.

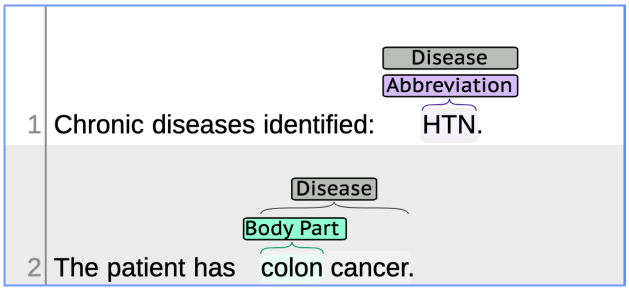


Figure 1.1: An example of a multi-label entity in our corpus, followed by a nesting of different types. The annotation was translated from its original language.

Regarding the second line of research, we have found two gaps in previous work on nested NER, which prevent a clean comparison between different approaches. First, we argue that the way the NLP community is evaluating the nested NER task does not adequately measure the effectiveness of a model at identifying nested entities, which is the main goal of the task. Specifically, the current metric calculates the micro F1-score by considering all entities in the partition separately, i.e., it does not distinguish between nested and flat entities. However, since flat entities are much more common than nested entities, the above metric ends up confusing flat and nested results and, consequently, is not able to reflect well the ability of a model to detect nesting. Second, although several approaches have been taken to deal with nested entities, we state that most of them rely on complex task-specific structures and ignore potentially useful baselines based on sequence labeling. We argue that this creates an overly optimistic impression of their performance.

That said, it is not clear whether we can achieve good performance on recognizing nested entities in our corpus by making simple modifications to sequence labeling-based architectures. On the other hand, considering the shortcomings in the area, it is interesting to study how generalizable these proposed models are, comparing them with other state-of-the-art architectures and testing them on other nested NER corpora.

1.2 Hypothesis

This work hypothesizes that it is possible to build a robust model for recognizing nested entities in the Chilean Waiting List corpus by using simple but powerful architectures based on sequence labeling. Besides, we believe that adding recent advances in deep learning, such as pre-trained language models, will give us even better results in finding as many entities as possible measuring with F1-Score value. Finally, we expect that these simple models will also have competitive performance compared to other state-of-the-art architectures and on different nested NER corpora.

1.3 Objectives

1.3.1 General Objective

The main objective of our research is to develop deep learning architectures to solve the nested NER task in our corpus, thus providing support for the Waiting List management through the secondary use of information. The idea is to establish which components of existing state-of-the-art architectures suit our problem and which ones do not. Applying and testing the proposed methods on other related corpora and comparing them with other state-of-the-art architectures is also part of the goal of this work.

1.3.2 Specific Objectives

1. Propose and develop deep neural architectures for solving the nested NER task in the Chilean Waiting List corpus.
2. Provide an empirical study comparing the proposed models with other state-of-the-art architectures in the nested NER task and testing these models on other related corpora to validate their effectiveness.
3. Introduce a formalization of the task by identifying the different types of nesting and then propose new task-specific evaluation metrics that adequately measure the model's performance on nesting.
4. Integrate the proposed models in a test environment, allowing health professionals to test them.

1.4 Methodology

In order to accomplish the specific objectives described above, this section presents the methodology proposed for our research. Precisely, our work mainly consists of the following steps:

1. Design a module to pre-process the annotations coming from the BRAT software and convert them to a format suitable for solving the flat NER and nested NER tasks.
2. Implement a baseline model following the flat NER approach, comparing it with different variants in its base architecture. The idea is to determine which components best suit our problem and which ones do not. For this purpose, different components commonly used in sequence labeling architectures will be tested, such as domain-specific word embeddings, character-level embeddings, LSTMs, transformers, and CRFs.
3. Characterize the occurrence of nested entities in our corpus, together with their clinical relevance and other characteristics that may be determinant when choosing an appropriate architecture.
4. Develop two deep neural architectures based on sequence labeling capable of addressing the nested NER task.
5. Replicate several state-of-the-art architectures in the nested NER task to study their performance on the Chilean Waiting List and compare their results with our best performing model.
6. Test the performance of our proposed model and baselines on other nested NER corpora from different domains and languages.
7. Implement new task-specific evaluation metrics that adequately measure the performance of these models on nested entities, which is the primary goal of nested NER.
8. Incorporate the best performing model into an existing web page, where through a simple interface, people in the medical field can test their performance on recognizing medical entities such as diseases, procedures, or clinical findings. Additionally, to speed up the annotation process by humans, this model would be included in a pre-annotation process.

1.5 Thesis Structure

The rest of the thesis is organized as follows:

In Chapter 2, we give a brief overview of the theoretical background needed to understand our research and a review of the related work in nested NER. Chapter 3 presents the data analysis and preliminary experiments following the flat NER approach. Next, Chapter 4 describes the deep neural architectures proposed to address the nested NER task in our corpus. In Chapter 5, we validate the effectiveness of the best-performing model by providing an empirical study comparing our approach with several state-of-the-art architectures and different corpora. Finally, the last chapter summarizes the conclusions of this work and discusses some of the future research lines for the project.

Chapter 2

Background and Related Work

This chapter reviews the scientific disciplines involved in our research and the related literature. First, it explains the technical background, briefly introducing the reader to the area of knowledge in which the thesis is developed and its methods. Then, it describes several corpora related to the clinical domain, such as ours. Finally, it presents a review of the different approaches proposed to handle the flat NER and nested NER tasks.

2.1 Scientific Disciplines

2.1.1 Artificial Intelligence

Artificial intelligence (AI) is a branch of computer science that seeks to develop algorithms capable of performing tasks that require human intelligence. As shown in Figure 2.1, there are three main fields in AI that aim to create systems or algorithms with intelligent behaviors: robotics, computer vision, and natural language processing.

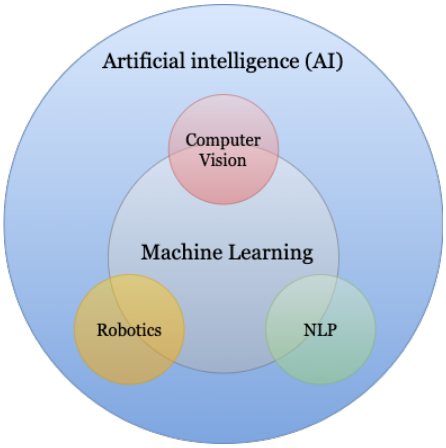


Figure 2.1: Diagram with the main disciplines belonging to the Artificial Intelligence field.

In recent years, several applications have been developed based on AI systems, such as

speech recognition systems, chatbots for customer service, recommendation engines, and image recognition applications. The development of these algorithms is based on three cognitive capabilities that allow them to be considered artificially intelligent: learning, reasoning, and adaptation. The subfield of AI that aims to integrate these three concepts through mathematical models is *Machine Learning*.

2.1.2 Machine Learning

Machine learning is an application of artificial intelligence that provides systems the ability to automatically learn and improve from experience without being explicitly programmed [101].

These algorithms are usually classified as *unsupervised* or *supervised models*. Unsupervised learning uses machine learning algorithms to analyze and cluster unlabeled datasets. In other words, it seeks to learn patterns within the data and group them according to those patterns. On the other hand, the supervised learning approach seeks to learn a mathematical function that connects the input data to an output prediction. The difference with the previous approach is that it requires a dataset composed of pairs of input and output data, better known as a labeled dataset. The word "supervised" comes from the fact that a human "supervisor" has previously categorized each input value with its corresponding output. In our work, we will use this approach since the Chilean Waiting List consists of labeled data.

Several methods have been proposed to tackle supervised learning problems, which can be divided into two main groups: classical machine learning and neural networks.

Classical Machine Learning

This approach consists of developing models that make certain assumptions about the data. This process is carried out through a complex feature engineering step, which means that human experts determine which features are best suited to understand the patterns between the input and output data. The main drawback of using this approach is the high cost in terms of money, time, and human resources. Later, we will discuss how this process has been sidelined due to recent advances in neural networks.

Neural Networks

One of the most widely used approaches to address supervised learning problems is using artificial neural networks [99], which have shown very positive results in many AI disciplines. The name comes from the shape of these architectures, which mimic the way the human brain works [84].

As shown in Figure 2.2, a neural network architecture is typically compound of hierarchical layers of neurons, where each layer processes certain information and propagates it to the next layer. This process is repeated until it reaches the final layer that produces the final output.

The first layer of a neural network is known as the input layer, the last layer is the output layer, and the layers in between are called hidden layers. Conventionally, a neural network is considered fully connected when each neuron in one layer is connected to all neurons in the next layer.

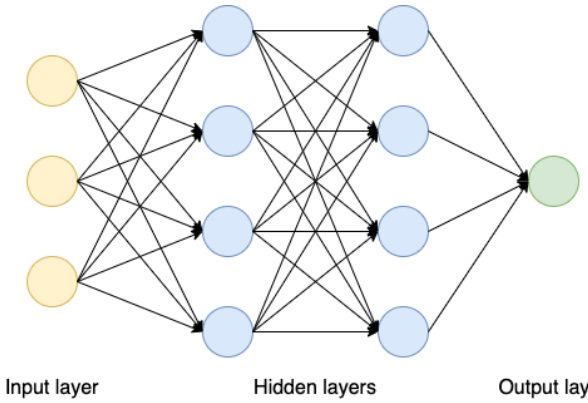
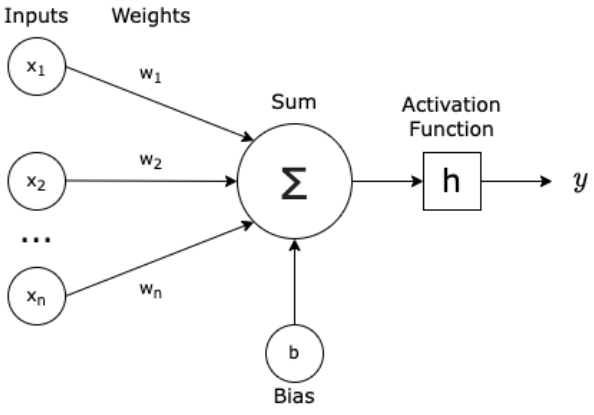


Figure 2.2: Fully connected artificial neural network.

Technically, artificial neurons can transmit information from the input to generate output through a series of mathematical operations. For example, Figure 2.3 shows a graphical representation of the simplest architecture of an artificial neural network, one that contains only one neuron. The inputs and output are numbers, and each input connection is associated with a weight. The unit (neuron) computes a weighted sum of its inputs and the bias, then applies an activation function to that sum and generates the final result. When neural networks have multiple neurons and layers, the information computed by one neuron is propagated to the neurons of the next layer, and so on. The particular arrangement and linking of these neurons allow the recognition of the underlying relations in a given dataset, which generates learning and allows the resolution of many computer application problems



$$y = h(x_1 \cdot w_1 + x_2 \cdot w_2 + \ldots + x_n \cdot w_n + b)$$

Figure 2.3: Diagram representing an artificial neuron.

Deep Learning

When neural networks grow considerably in size, i.e., the number of layers and neurons, we reach the field of Deep Learning (DL). Formally, it describes a family of learning algorithms rather than a single method that can be used to learn complex prediction models by using multi-layer neural networks with many hidden units [66]. The main advantage of using these large models is that they can learn highly complex patterns, leaving feature engineering aside and learning these representations by the algorithms themselves.

In recent years, these algorithms have become more popular due, in part, to improvements in hardware and computational capacity, the increased availability of data to train the models, and advances in the field of machine learning [27]. Classic examples of these architectures are Recurrent Neural Networks (RNN) [100], Convolutional Neural Networks (CNN) [65], and Transformers [118]. These models are potentially useful when working with unstructured data, such as audio, images, videos, and text, which is the focus area of this thesis.

2.1.3 Natural Language Processing

The analysis of unstructured texts written by humans is challenging since it is complex to formally understand and describe the rules governing human language, as it is very ambiguous and constantly evolving.

Natural Language Processing (NLP) is an interdisciplinary field of artificial intelligence that involves computer science and linguistics disciplines. NLP aims to develop algorithms capable of understanding, interpreting, and manipulating natural human language. Precisely, it seeks to develop computational systems used for solving practical problems involving human language. These problems are better known as *NLP tasks*, which can be divided into three main groups:

- Text Classification: This task aims to classify documents into predefined categories, typically using machine learning algorithms. These systems can be used to organize, structure, and categorize unstructured text based on its context. Typical applications belonging to this category are sentiment analysis, spam filtering, language detection, and hate speech recognition.
- Sequence Labeling: This task aims to assign a class or label to each token in a given input sequence. These labels are useful to create statistics about the data, summarize key information, and in other cases, are used as features in downstream models. Classic examples of sequence labeling problems are part-of-speech tagging (POS), word sense disambiguation, word segmentation, and named entity recognition (NER).
- Sequence to Sequence: This task aims to map a fixed-length input with a fixed-length output, where the length of the input and output may differ [111]. It is commonly used in sequence prediction tasks, such as language modeling and machine translation. Practical problems associated with this task are chatbots, language translators, summarization, question answering, or any application that generates new sequences of natural language texts.

Several computational methods have been proposed to address these natural human language tasks, which can be divided into three main approaches:

- Rule-based Systems: This approach consists of designing hand-crafted rules to incorporate knowledge and reasoning mechanisms into intelligent NLP systems. In simple words, the aim is to find linguistic rules, patterns, or regularities in data that can be expressed using "IF-ELSE" statements. The major drawback of using this method lies in the inability to model larger corpora rules optimally, the difficulty of their maintenance, and the requirement of skilled developers and linguists to manually encode each rule.
- Classical Machine Learning: As explained in section 2.1.2, another classic approach in NLP is to use expert knowledge to determine which are the best features associated with the model's input sentences. Here, the algorithm starts analyzing the corpus and features to produce its own rules, classifiers, and knowledge. The most commonly used algorithms are Naive Bayes, Hidden Markov Models, and Support Vector Machines. The creation of these models is simpler than the previous approach, achieving better performance and also speeding up the development of NLP systems. However, the main limitation with this approach is the lack of training data, which requires a great deal of human effort to build the corpora. Additionally, it is not an end-to-end system since most of these systems are accompanied by complex feature engineering.
- Deep Learning: Currently, the best results in NLP tasks have been obtained using deep learning-based architectures. Under this approach, the most commonly used models are recurrent neural networks, convolutional neural networks, encoder-decoder architectures, and transformers. In addition, the representation of words into numerical representations is usually performed by using domain-specific word embeddings, character-level embeddings, and contextual word embeddings. This means that an expert is no longer required to encode rules or features by hand, as in the previous two appoaches.

The following sections describe the NLP task addressed in our research, which is called *Named Entity Recognition (NER)*. This problem belongs to the sequence labeling category and is commonly addressed using deep learning techniques.

2.2 Named Entity Recognition

Named Entity Recognition (NER) is an important task in NLP that seeks to identify sequences of words (entities) expressing references to predefined categories (entity types). NER, or in general the task of recognizing entity mentions[1], has drawn the attention of the research community due to its relevance in several NLP applications such as relation extraction [88], entity linking [40] and co-reference resolution [19].

In early work, NER was used to identify personal names, organizations, and locations [20]. For example, in Figure 2.4, four different entity types are identified: personal names,

[1]Mentions are defined as references to entities that could be named, nominal or pronominal [36].

organization, numbers, and dates. However, entities have been extended to various domains and applications in recent years, such as the example of our clinical corpus.

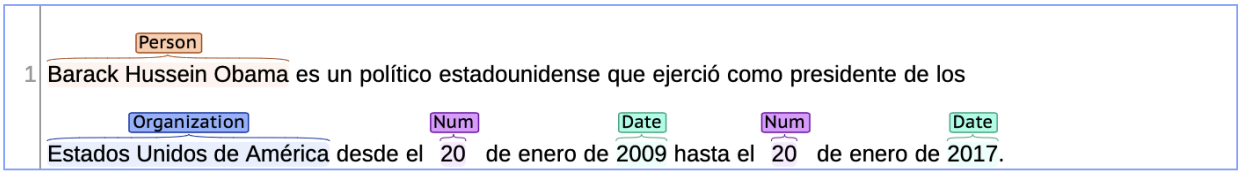


Figure 2.4: Example of named entities extracted using the Stanford NER system [34].

Task Formalization

In most NLP tasks, a formal definition is usually introduced in order to understand the problem better. This process consists of identifying the input and output variables of the task under study. In our context, we present below a definition proposed by us for the NER task.

Definition 1 (NER) Given an input sequence $X = \{x_1, x_2, ..., x_n\}$ of words, an entity Q is defined by a tuple (S_q, E_q, T_q), where S_q and $E_q \in [1, n]$ represents entity boundaries in X, and T_q in $\mathcal{E}$ (the entity space) corresponds to entity type. The aim of NER is to correctly identify the boundaries for every entity Q in X and assign it the correct entity type from a predefined list of categories.

Evaluation Metrics

Once the task has been formally defined, it is important to establish which evaluation metrics will be used to compare the predictions of NER systems against real labels.

The official NER metric was proposed in the CoNLL-03 conference [102] and consists of calculating the micro F1-score using a strict evaluation approach. This metric considers an entity correct when both entity types and boundaries are predicted correctly. Most of the studies use micro over macro measurement when there is an imbalance of possible classes, and it is necessary to weigh the results according to the frequencies of each class. Below, we describe each of the concepts needed to calculate the F1 measure.

Precision (P), which is also known as the positive predictive value, is computed based on the count of true positives (TP) and false positives (FP). Intuitively, this metric calculates which percentage of named entities found by the NER system is present in the real labels.

$$P = \frac{TP}{TP + FP}$$

Recall (R), which is also called sensitivity, is calculated out from the number of true positives (TP) and false negatives (FN). Intuitively, this metric calculates which percentage

of named entities present in the real labels is found by the NER system.

$$R = \frac{TP}{TP + FN}$$

Finally, the *F1-score (F1)* is the harmonic mean of precision and recall scores, reaching its best value at 1 (perfect precision and recall) and worst at 0:

$$F1 = \frac{2 \cdot P \cdot R}{P + R}$$

2.2.1 Nested Named Entity Recognition

Nested Named Entity Recognition is a particular case of NER where entities are nested within each other [33]. In Figure 2.5 we show an example in the Chilean Waiting List corpus. For instance, the entity "ovarios con 2 quistes" is a Disease containing "ovarios", which is a Body Part.

Figure 2.5: An example of an annotation in the Chilean Waiting List corpus, which contains nested entities.

Traditional NER systems simplify nested entities by keeping the outermost entity and eliminating the inner ones. This simplified problem is better known as flat NER and is commonly regarded as a sequence labeling task ([61], [79], [97]). Under this approach, the main assumption is that each token can be associated with at most one label, thus ignoring nested entities.

In the clinical domain, it is more common to see flat NER solutions in publications. However, the nested NER task is more complex and challenging because of the relations between entities in this field. Therefore, removing part of these entities could be a problem in model performance due to losing relevant clinical information.

Task Formalization

One of the main issues in our knowledge of nested NER is that the task definition has not been addressed in-depth, and clarification of the different nesting cases is needed. After analyzing three corpora containing nested entities, we have identified the following nesting cases:

- **Multi-label entities (ME)**: This case has been little explored in the literature. As explained in Alex et al. [5], it consists of entities tagged with more than one entity type. With the release of the Chilean Waiting List corpus, it is interesting to study this case since 10.75% of the entities are involved in this type of nesting. For example, the entity "HTN", which stands for hypertension, is tagged as a disease and an abbreviation.
- **Nested entities of different types (NDT)**: This is the most frequent type of nesting in nested NER datasets. It consists of an entity containing a shorter entity tagged with a different type. An example is "colon cancer", where a body part (colon) is contained in a disease.
- **Nested entities of the same type (NST)**: This case usually occurs when entities are originally represented by a hierarchy, which is later pruned to reduce the entity space, resulting in the merging of entities of different levels of granularity. Although it appears in most corpora, it is much more frequent in GENIA [52]. For example, the DNA "Drosophila homeodomain" contains another DNA, "homeodomain".

To better understand these cases, we formally define what we mean by nested entities and the nested NER task.

Definition 2 (Nested entities) Given an input sequence $X = \{x_1, x_2, ..., x_n\}$ of words, an entity Q is defined by a tuple (S_q, E_q, T_q), where S_q and $E_q \in [1, n]$ represents entity boundaries in X, and T_q in $\mathcal{E}$ (the entity space) corresponds to entity type. Given two entities Q and R, we say that Q is nested in R if $S_r \leq S_q$ and $E_q \leq E_r$. The particular case of $S_q = S_r$ and $E_q = E_r$ corresponds to an entity with multiple labels. Note that under this definition we consider the three types of nesting described above.

Definition 3 (Nested NER) Given an input sequence $X = \{x_1, x_2, ..., x_n\}$, nested NER aims to correctly identify the boundaries for every entity Q in X and assign it the correct entity type from a predefined list of categories. This identification must be made for cases where nested entities are involved and when not.

To the best of our knowledge, this is the first effort to provide a formal definition of the nested NER task. This contribution may improve future work by paying particular attention to the different nesting cases described. For example, deciding which model to use in a given dataset could be based on the model's ability to identify these cases.

Evaluation Metrics

Regarding the evaluation of nested NER systems, the standard metric used to compare different approaches is the one described for the flat NER case but considering all entities in the test partition. However, since entities in nested NER corpora are mostly not nested, we argue that this metric may not adequately measure the performance of models on nesting. Using the above metric implies that models that recognize flat entities well will have a high metric even though they do not recognize nested entities correctly. Given these facts, proposing new evaluation metrics for this task is another part of our work.

2.3 Related Work

This section describes annotated corpora related to the Spanish clinical domain, such as ours. Then, it summarizes the methods used by different authors to address the flat NER task. Finally, it reviews the current state-of-the-art models used to address the nested NER task, which is the primary goal of this research.

2.3.1 Annotated corpora

Machine understanding of clinical texts requires dealing with a non-standardized use of the language, mainly due to the heavy use of abbreviations, local jargon, and significant spelling errors. Because of this, there is a need to build annotated corpora that allow the development of models that can address these challenges automatically.

Although Spanish is the fourth most spoken language globally, there is still a lack of annotated resources. In terms of linguistic resources using clinical text in Spanish, publications from Spain are predominant, such as the work by Oronoz et al. [94] that annotated diseases, drugs, and substances in medical records. The same group published a corpus afterward for adverse drug reactions [95]. From Spanish-speaking countries besides Spain, and to the best of our knowledge, the only published work is by Cotik et al. [22] in Argentina for the annotation of clinical findings, body parts, negation, temporal terms, and abbreviations in radiology reports. These works inspired part of the creation of our corpus since they annotated similar entities.

Some of the work done on Spanish biomedical texts is also noteworthy; Moreno-Sandoval and Campillos-Llanos [90] annotated Part-of-Speech in biomedical documents written in Spanish, Japanese, and Arabic, and Krallinger et al. [56] annotated PubMed abstracts in Spanish with chemicals and drugs. Several works have created resources in Spanish for entity recognition and clinical coding to internationally recognized classification systems; Kors et al. [55] created a multilingual corpus for biomedical concept recognition, Campillos-Llanos [15] created a medical lexicon and a clinical trials corpus [16] with words and entities mapped to the Unified Medical Language System (UMLS) [72] identifiers, while Intxaurrondo et al. [47] manually annotated abbreviation mentions and their definitions from clinical case studies and mapped them to control vocabulary resources such as the Systematized Nomenclature of

Medicine – Clinical Terms (SNOMED-CT). Finally, there is the work of Miranda-Escalada et al. [89] who published resources and methods for automatic clinical coding to the International Statistical Classification of diseases and Related Health Problems (ICD-10) on medical documents.

Regarding the nested NER task, there is a scarcity of annotated resources. The closest work is an English biomedical corpus called GENIA [52], which was obtained from thousands of MEDLINE abstracts. Later, we will describe this resource in-depth since it has served as inspiration for many nested NER works.

2.3.2 Named Entity Recognition

Named Entity Recognition (NER) has been studied for decades by the NLP research community. In early work, the entity types had a more general-purpose, such as locations, person names, and organizations. Nowadays, we can see named entities belonging to varied domains and applications, such as the case of our clinical corpus. Regarding the methods proposed to recognize entities, we can categorize the related work into three main groups: rule-based methods, classical machine learning models, and neural networks.

Rule-based Systems

This approach is strongly related to the design of hand-crafted rules based on semantic and syntactic patterns. In most cases, these rules tend to have the form of *IF-ELSE* statements. Some examples of rule-based NER systems include LaSIE-II [46], FASTUS [42], and LTG [85]. Although this approach seems to be very simple and effective, the problem lies in the scalability of these models on huge and complex text corpora. Since there are large volumes of data nowadays, it has become infeasible to continue developing these systems for the NER task.

Classical Machine Learning

This approach aims to develop supervised NER systems, in which, unlike more recent architectures, feature engineering is a fundamental building block. Most of the work is based on the design of reliable word-level features, such as morphology and part-of-speech tags. Then, based on these features, many machine learning algorithms have been proposed. Classic examples are hidden markov models (HMMs), decision trees, maximum entropy models, support vector machines (SVMs), and conditional random fields (CRFs).

In Bikel et al. [14], they proposed the first HMM-based NER system, named IdentiFinder. This model was implemented to identify and classify names, dates, time expressions, and numerical quantities. Another work by Szarvas et al. [112] developed a multilingual NER system by using the C4.5 decision tree and AdaBoostM1 learning algorithm. In addition, CRF-based systems have been widely used in NER, even in the most modern architectures. Early work from Kim et al. [51] proposed a feature induction method for CRFs in NER.

Similarly, Krishnan and Manning [57] proposed a two-stage approach based on two coupled CRF classifiers. The second CRF makes use of the latent representations derived from the output of the first CRF. As mentioned above, the problem with these methods is that they rely on human feature engineering, which is not optimal in terms of time, and resources.

Neural Networks

In recent years, neural networks have proven to build reliable NER systems without handcrafted features or task-specific knowledge. Most of the existing work formulates NER as a sequence labeling problem, which makes the central assumption that each token is tagged with at most one label. This approach is better known as flat NER and does not consider nested entities. The analysis of flat NER models is generally divided into three main layers: the embedding, encoder, and classification layers.

Representing words into numerical vectors has proven to be a fundamental building block when constructing neural network architectures. The most traditional representation is word embeddings, a vector representation that allows words with similar meanings to have a similar representation. Along with these embeddings, it is common to concatenate embeddings at the character level to enhance the representation of rare and out-of-vocabulary words. These embeddings are usually generated by using a LSTM [61], or CNN [79].

With recent advances in deep neural networks, there are more robust token representations retrieved from pre-trained language models, such as BERT [29], Flair [3] and LUKE [126]. This type of representation has made it possible to achieve the state-of-the-art in the flat NER task, for example, with Flair-based architectures. In our experiments, we leverage these contextualized embeddings to obtain a significant improvement in the model's performance.

Several techniques have been proposed in the literature regarding the classification layer, which seeks to transform the representations obtained in previous layers to their respective categories. The main one is the linear chain CRF [59], which obtains the most probable sequence of labels associated with the input. This method has reached the state of the art in several articles ([79], [61], [3], [122]).

2.3.3 Nested Named Entity Recognition

As mentioned above, the problem of using the flat NER approach is to assume that a token can be tagged with a single label, which does not allow the appearence of nested entities. Therefore, it is necessary to propose models that are capable of dealing with these types of entities.

The first solutions that attempted to predict nested structures used a combination of Hidden Markov Models (HMM) to detect subsets of named entities and handcrafted rules to expand these subsets [104, 129, 133]. Support vector machines (SVM) have also been used to identify nested entities. Zhou [132] combined such a model with a rule-based approach, while Gu [39] used two separate SVM models to detect the innermost and outermost entities.

In the last years, there has been a growing interest from the research community in designing neural models to address the nested NER task. Several studies have been conducted, which can be mainly divided into three categories: region-based, structure-based, and sequence labeling-based.

Region-based

These approaches divide the problem into two sequential stages: identifying entity boundaries and then categorizing these regions. In Sohrab and Miwa [107], they designed a model that enumerates all possible spans within a limited length. The entity types are predicted by using boundary and average internal token representation. Another region-based model was proposed by Zheng et al. [131], which uses a sequence labeling layer to detect entity boundaries, and then classifies selected regions into their categorical types. In recent work by Yu et al. [127], they used ideas from a biaffine model, scoring all possible start-end tokens in a sentence to predict nested entities.

Although these methods have proven effective, they often suffer from high time complexity, fail to capture the interaction between outermost and inner entities, and cannot identify entities tagged with more than one entity type, a frequent nesting type in our corpus.

Structure-based

There have also been attempts to capture the structure of nested entities. In other words, the aim is to create data structures capable of finding the relations between inner and outermost entities. Finkel and Manning [33] represented each input sentence as a constituency tree of nested entities and used a CRF-based approach to predict entity types. Lu and Roth [76] proposed a mention hypergraph representation to extract entity mentions. Next, Muis and Lu [91] improved on previous work by modeling nested NER with mention separators and handcrafted features. However, their method requires multiple graphs if there is more than one entity type. Similarly, Katiyar and Cardie [50] designed a directed hypergraph using LSTM features to learn the nesting structure. Finally, Wang et al. [123] recursively introduce the embedding of tokens and regions into flat NER layers simulating the shape of a pyramid and extracting nested entities from the innermost to the outermost entities. This method is precisely the state of the art in nested NER.

Structure-based approaches are capable of modeling proprietary structures to explicitly capture nested entities. However, although this approach has achieved good performance on nested NER, most of them need extra annotation, complex feature engineering, or suffer from spurious structures and structural ambiguities, as explained in Wang and Lu [120].

Sequence labeling-based

Some authors state that sequence labeling methods can also be adapted and perform well on this task. This approach transforms the nested NER task into a special sequential labeling

task by designing a suitable tagging schema.

Early work mainly exploited the potential of conditional random fields (CRF). In Alex et al. [5], they proposed three CRF-based methods to reduce the nested NER as several BIO tagging problems. Their best approach, called cascaded CRF, uses one model per entity type using the output of the previous flat NER model as a feature for the current one. However, this approach cannot handle nested entities of the same entity type because type-specific CRF models generate flat predictions. Ju et al. [49] took advantage of inner entity information to encourage outer entity recognition. They dynamically stacked LSTM-CRF layers predicting entities in an inside-to-outside way until no entities were extracted. Although this method can deal with nested entities of the same type, it suffers from error propagation from lower to higher layers. The wrong entities extracted by the previous layer will affect the recognition performance in the next layer.

Straková et al. [110] proposed modeling nested NER in two ways: First, using a sequence labeling approach by concatenating multiple labels into one single label. Second, treating nested NER as a sequence-to-sequence problem using an LSTM to decode entity types. Finally, Shibuya and Hovy [106] recognized entities iteratively from outermost ones to inner ones using a recursive method based on CRFs. As a preview, in Table 2.1, we can see the main results obtained by some of the mentioned architectures on two nested NER datasets, which will be used to test the effectiveness of our final model and our datasets.

	GENIA			**GermEval**		
Model	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
Ju et al. [49]	73.9	68.7	71.2	71.8	64.1	67.7
Sohrab and Miwa [107]	74.1	69.7	71.8	78.6	64.6	70.9
Zheng et al. [131]	76.7	71.8	74.2	74.4	65.5	69.7
Wang et al. [124]	-	-	-	74.8	70.5	72.6
Wang et al. [123]	78.1	72.8	75.3	77.8	66.9	71.9
Yu et al. [127]	79.1	73.7	76.3	89.0	77.4	82.8
Shibuya and Hovy [106]	75.8	75.2	75.5	85.1	78.2	81.5
LM-based						
Dadas and Protasiewicz [24] [BERT + Flair]	-	-	-	86.6	80.6	83.5
Luan et al. [77] [ELMO]	-	-	76.2	-	-	-
Straková et al. [110] [BERT + Flair]	-	-	78.3	-	-	-
Wang et al. [123] [BERT + Flair]	80.3	78.3	79.3	-	-	-
Yu et al. [127] [BERT]	79.9	76.5	78.1	88.3	85.0	86.6
Shibuya and Hovy [106] [Flair]	77.1	78.0	77.6	83.4	82.9	83.2
Wang et al. [123] [BERT]	79.1	76.9	78.0	87.7	85.8	86.7

Table 2.1: Overall results of the revisited models on two nested NER corpora.

In this research, we argue that the NLP community has little explored the sequence labeling-based approach despite its effectiveness. Precisely, we found some simple but overlooked sequence labeling-based models with a competitive performance compared to more sophisticated methods specifically designed to address this task.

Chapter 3

Flat Named Entity Recognition in the Chilean Waiting List

This chapter describes preliminary NER experiments performed in the Chilean Waiting List corpus. First, it explains the steps needed to pre-process the annotations provided by the research group, together with a detailed analysis of the data. Then, several components commonly used in sequence labeling architectures are tested, following the flat NER approach, i.e., without considering nested entities. The aim is to provide an initial exploration of the dataset and establish which components best suit our problem and which ones do not. This information will be helpful to design our nested NER architectures and establish the importance of considering nested entities in our corpus.

3.1 Data Description

This section describes how the annotation process was previously carried out. Then, we describe the steps needed to transform the annotations to the standard NER format. Finally, we provide a detailed description of the data and the challenges in our corpus.

3.1.1 Annotation Process

In 2018, the group requested the non-GES Waiting List from the 29 health services in the country through Transparency Law. These requests were answered positively by 23 of the health services and sent WL datasets for years between 2008 and 2018. Considering only the reasons for referral, we collected 994,946 different diagnoses.

A random subset of these diagnoses was selected for annotation, with the criterion of selecting those with more than 100 characters. Using this condition, we reduce the corpus to 107,235 unique candidates. Moreover, we removed diagnoses with text imperfections. After filtering, one of the managers inspected each remaining diagnosis to ensure that they fully met the conditions. Even though the referrals come de-identified from the source, this person

also checked for any personal information.

Having these referrals, the purpose of the annotation process was to build a corpus with a considerable volume of labeled text to train NER models. Specifically, this process consisted of using expert knowledge to identify pieces of text with medical relevance. For this purpose, the manual annotation of the referrals was done using the BRAT annotation software, a web-based tool for adding notes to existing text documents [109]. This platform offers an intuitive user interface, flexible configuration of the annotation scheme, and workflow support for annotation stages.

The annotation process involved three stages. Figure 3.1 illustrates the process used to create the first 900 annotations, but the rest of the annotations have been consolidated in the same way to date. Here, four annotators (three medical students and one medical doctor) were selected for the initial stage, who were permanently supported by three project managers. To improve the quality of the annotations, the clinical experts followed a strict annotation guide, which is freely available here[1].

A test version of the annotation guidelines was written in the first stage. These guidelines were evaluated during the annotation of 25 referrals. In the second stage, the three medical students annotated 50 identical referrals in weekly annotation rounds for three weeks. In an iterative improvement process, the medical students were retrained after each round of annotation. At this point, the guidelines were further modified to clarify the task and improve consistency. At the end of this stage, the first accepted version of the guidelines was established and released. In stage three, a medical doctor joined the group (namely a senior annotator) and was asked to annotate the same 150 referrals done by the students independently. Finally, for the consolidation process, we decided to have each annotation revised by a team of four researchers, including the senior annotator, a dentist, the postdoc that created the annotation guidelines, and the principal investigator.

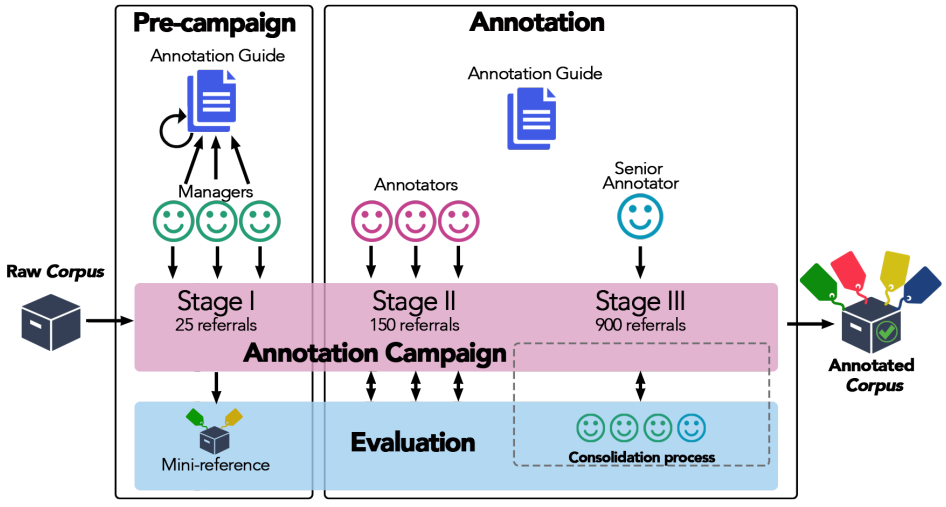


Figure 3.1: Annotation stages for the creation of annotation guidelines, the training of the senior annotator, and the production stage where referrals were consolidated.

In summary, Figure 3.2 shows all the entity types agreed upon by the research team. In

[1] https://plncmm.github.io/annodoc/

the case of clinical findings and procedures, we only used the parent entities of the hierarchy, leaving a total of seven entity types. The choice of these categories was based on literature revision and our interest in the Waiting List. For example, we were interested in describing how many procedures were pending or mining the family history of diseases. A brief description of each entity type is presented below the figure.

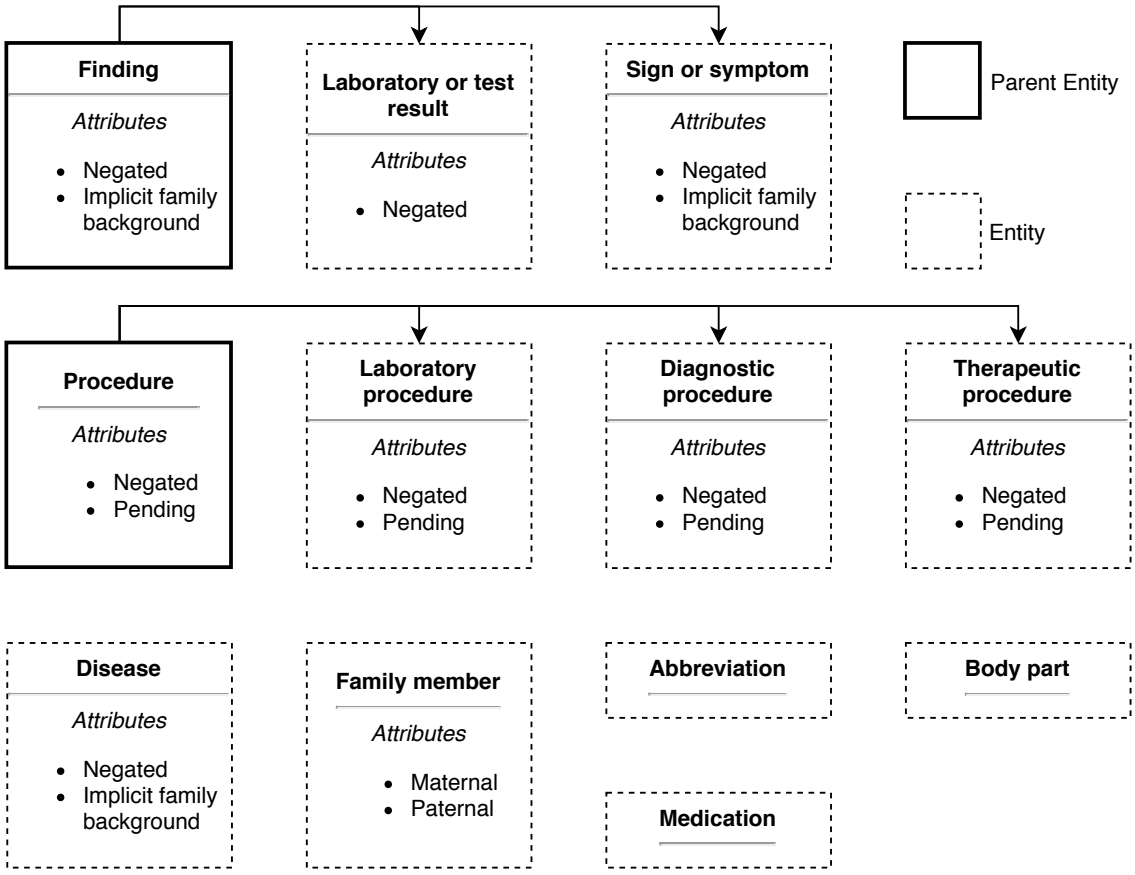


Figure 3.2: List of entity types (in bold) in the Chilean Waiting List.

- Abbreviation: Linguistic procedure to shorten the morphology of certain words.
- Body Part: An organ or an anatomical part of a person.
- Disease: An alteration or deviation of the physiological state in one or more parts of the body due to generally known causes. These causes manifest themselves with characteristic symptoms and signs, whose evolution is more or less predictable.
- Finding: Observations, judgments, or evaluations made about patients.
- Procedure: Activities derived from patient care and attention.
- Medication: Mentions of medicines or drugs used in the treatment and prevention of diseases, including brand names and generics, as well as names for groups of medicines.
- Family Member: Consanguineous and non-consanguineous relatives mentioned in the diagnoses.

To date, our research group has consolidated 5,000 annotations of non-GES list referrals. Although the annotation process is still ongoing, we will work with this fixed number of annotations to be consistent in our experiments. The Chilean Waiting List corpus is freely available for non-commercial use[2].

Data Preprocessing

Figure 3.3 shows a medical annotation using the BRAT platform. We will use this example to show how to convert these annotations into a standard flat NER format.

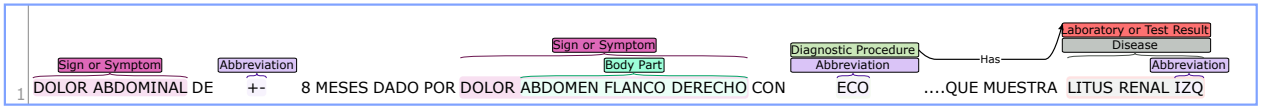


Figure 3.3: Text fragment of a referral annotated with the BRAT software.

Initially, the BRAT annotation platform generates a file in a format called *standoff*. As shown in Figure 3.4, this format follows a basic structure with columns containing an ID per annotation, the entity type, the indices of start-end characters of the annotation, and the string that constitutes the entity. Since it is uncommon to use this format to represent corpora in the NER task, we implemented a pre-processing tool[3] to transform the standoff files to the *CoNLL* format [115], which corresponds to the standard input representation for training NER models.

```
T1  Finding 0 15    DOLOR ABDOMINAL
T2  Abbreviation 19 20  +-
T3  Finding 38 66   DOLOR ABDOMEN FLANCO DERECHO
T4  Body_Part 56 66 ABDOMEN FLANCO DERECHO
T5  Procedure 71 74 ECO
T6  Abbreviation 71 74 ECO
T7  Finding 87 102 LITUS RENAL IZQ
T8  Disease 87 102 LITUS RENAL IZQ
T9  Abbreviation 99 102 IZQ
```

Figure 3.4: Annotation of Figure 3.3, transformed to the standoff file format.

Figure 3.5 shows the general structure of CoNLL files. This format consists of two columns: one column contains the tokens, and the other contains their associated labels. To separate one sentence from another, a blank line is used. Unfortunately, this format does not support nested entities since each token is associated with at most one label. Therefore, we have to choose which nested entity to use. In this preliminary experiments, and for the sake of simplicity, we kept only the outermost entities that compose a nesting, thus facilitating the creation of the CoNLL files. When a span is annotated with more than one entity type, we arbitrarily keep one of them.

[2] https://zenodo.org/record/5518225
[3] https://github.com/plncmm/acm_health_msen

```
DOLOR   B-Finding
ABDOMINAL   I-Finding
DE  O
+-  B-Abbreviation
8   O
MESES   O
DADO    O
POR O
DOLOR B-Finding
ABDOMEN I-Finding
FLANCO  I-Finding
DERECHO I-Finding
CON O
ECO B-Procedure
QUE O
MUESTRA O
LITUS   B-Disease
RENAL   I-Disease
IZQ I-Disease
```

Figure 3.5: Annotation of Figure 3.4, transformed to the CoNLL file format.

To convert sentences into sequences of tokens, we used the *esnewslg* model, which is a Spanish statistical tokenizer available at the Spacy library [45]. This model was trained using the Spanish AnCora and WikiNER datasets. In addition, to handle misspellings and out-of-vocabulary words, we added a second tokenizer based on regular expressions. The labels were encoded following the standard *IOB2* format, which assigns a label to a token depending on its position in the entity found. The *B-* prefix is assigned to the tokens located at the beginning of an entity, and the *I-* prefix when the token is within an entity. If the token does not belong to any entity type, we use the *O* label.

3.1.2 Data Exploration

Data exploration is one of the most critical steps when designing NLP systems. This process allows us to find relations in the data that could be an important support when developing NER models. As previously mentioned, to perform an initial exploration of the corpus, we followed the traditional flat NER approach, in which nested entities are not considered. Later, we will discuss how these results varied when considering the deleted nested entities.

The corpus is a collection of 5,000 referrals divided into 2,067 dental and 2,933 medical. The documents distribution among the dental and medical specialties are described in Tables 3.1 and 3.2, respectively. It is interesting the large number of specialties involved in these referrals, which means that there will be greater variability in the terms and abbreviations used in these texts.

Table 3.3 shows the overall statistics of our corpus. After preprocessing 5,000 referrals, we obtained 9,894 sentences. 8,014 were used for training, 890 for validation, and 990 for testing, leading to a ratio of 0.81: 0.09: 0.1, the same ratio used in GENIA [52], which is the most similar corpus in nested NER. These partitions can be found in the official repository

Dental specialty	Documents	Percentage
Oral rehabilitation: Removable dentures	515	10.30 %
Endodontics	501	10.02 %
Orthodontics	343	6.86 %
Periodontology	343	6.86 %
Maxillofacial surgery	142	2.84 %
Oral Surgery	114	2.28 %
Oral rehabilitation: Crowns	51	1.02 %
Operative dentistry	23	0.46 %
Temporomandibular disorders and orofacial pain	3	0.06 %
General dentistry	3	0.06 %

Table 3.1: Documents distribution by dental specialty.

	Documents	Percentage
Traumatology	489	9.78 %
Gynecology	277	5.54 %
Otorhinolaryngology	223	4.46 %
Ophthalmology	216	4.32 %
Neurology	197	3.94 %
Internal medicine	174	3.48 %
Surgery	168	3.36 %
Pediatrics	158	3.16 %
Cardiology	150	3.00 %
Gastroenterology	131	2.62 %
Dermatology	105	2.10 %
Urology	96	1.92 %
Psychiatry	80	1.60 %
Vascular surgery	64	1.28 %
Endocrinology	56	1.12 %
Pediatric surgery	53	1.06 %
Nephrology	53	1.06 %
Pulmonology	43	0.86 %
Obstetrics	43	0.86 %
Neurosurgery	38	0.76 %
Abdominal surgery	23	0.46 %
Rheumatology	20	0.40 %
Hematology	15	0.3 %
Physical medicine and rehabilitation	13	0.26 %
Infectology	10	0.20 %
Oncology	9	0.18 %
Genetics	9	0.18 %
Colorectal surgery	7	0.14 %
Breast Surgery	6	0.12 %
Plastic Surgery	3	0.06 %
Geriatrics	2	0.04 %
Cardiothoracic Surgery	1	0.02 %
Anesthesiology	1	0.02 %

Table 3.2: Documents distribution by medical specialty.

of our corpus.

Concerning the number of tokens, we have about four times fewer data than other nested NER datasets, such as GENIA [52] or GermEval [12], which could affect the performance of deep learning models. In terms of the annotated tokens, we observe that more than 50% of the tokens are associated with some entity in each partition, proving to be an excellent NER dataset. Regarding the average number of tokens and entities per sentence, we can see that the numbers are similar between different partitions, which provides more reliability on the data distribution obtained at the time of partitioning.

The chart presented in Figure 3.6 shows the frequency of entity types in our corpus. These values are calculated after cleaning the nested entities, i.e., leaving only the outermost entities in each nesting. First, we observe an imbalance in the classes, where the most frequent entity types correspond to findings and diseases. According to experts, these entity types are also

Metric	Train	Dev	Test
Sentences	8,014	890	990
Entities	26,391	2,949	3,192
Tokens	149,574	16,754	18,436
Annotated tokens	78,050	8,557	9,804
Vocabulary	17,421	4,281	4,680
Lexical diversity	11.6%	25.6%	25.4%
Mean tokens per sentence	18.7	18.8	18.6
Mean tokens per entity	2.96	2.90	3.07
Mean entities per sentence	3.29	3.31	3.22

Table 3.3: Statistics of the Chilean Waiting List corpus without considering nested entities.

the most difficult to recognize manually. Besides, they state that the easiest categories to recognize are medications and family members, the least frequent in the chart. For this reason, it is important to choose an appropriate evaluation metric capable of handling this class imbalance.

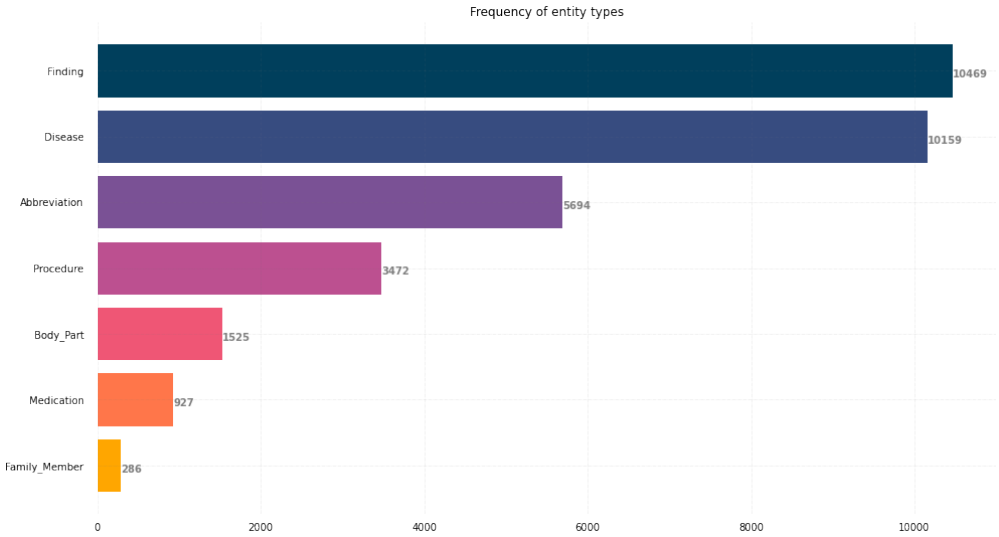


Figure 3.6: Frequency of entity types in our corpus without considering nested entities.

In Figure 3.7, we study the distribution of the lengths for each entity type. We can observe that the most frequent entities are also the longest ones, which makes recognizing these entities an even more complex task. The main reason is the strict evaluation metric employed, where an entity is considered correct when both the entity type and the boundaries coincide. Thus, having very long entities makes the model more vulnerable to making errors in the boundaries. Finally, we observe that Abbreviation is the entity type with the highest average number of occurrences in the annotations, which evidence the difficulty of manual analysis of these texts.

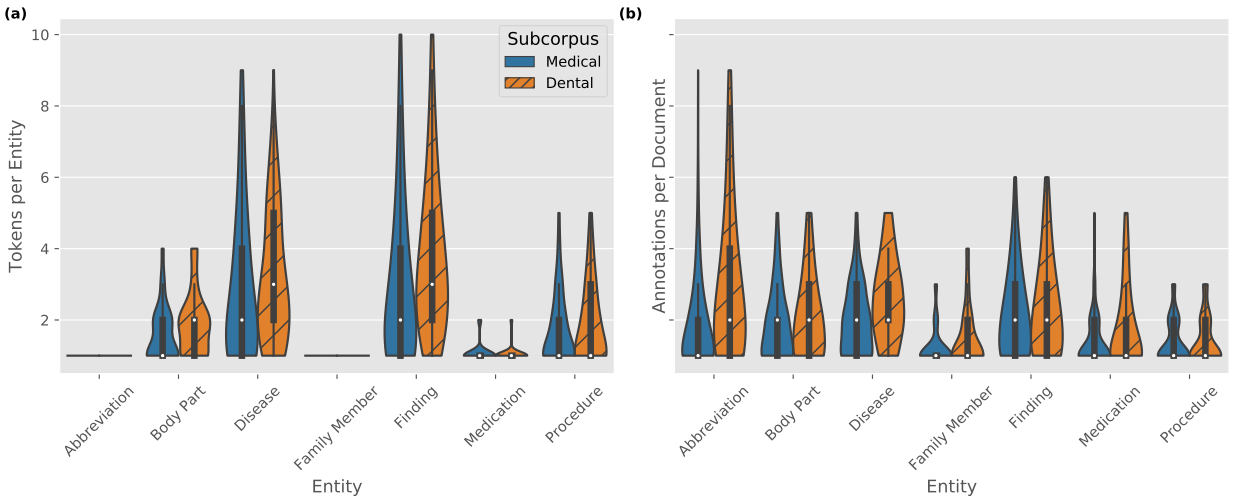


Figure 3.7: Frequency distribution and median (white point) of (a) tokens per entity across the subcorpus, and (b) annotated entities per document by subcorpus.

3.2 Methods

In this section, we present preliminary experiments performed in the Chilean Waiting List corpus. Specifically, we tested different sequence labeling-based architectures, which have shown outstanding performance in the flat NER task. For ease of reading, Figure 3.8 shows the elements studied according to the following layers of the neural network: embedding, encoder, and classification layers.

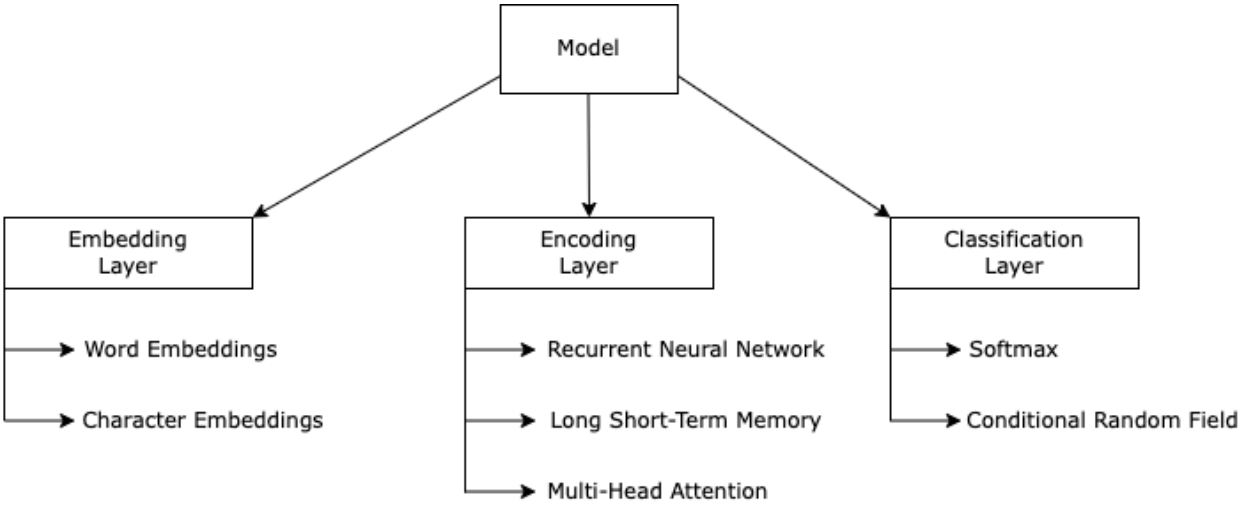


Figure 3.8: Diagram with the different architectures tested in our experiments.

3.2.1 Embedding Layer

In order to create an architecture based on neural networks, the first step is to represent words as mathematical structures, thus allowing a computer to operate on them. In the following lines, we describe the different numerical representations chosen in our experiments.

Word Embeddings

Word embedding is one of the most popular and efficient ways to convert words into numerical vectors. It is capable of capturing the semantics of words in dense, low-dimensional continuous vectors. This process allows words with similar meanings to have a similar vector in the embeddings space, which is better known as the distributional hypothesis.

There are two main approaches used to incorporate word embeddings into deep learning models. The first method consists of adding an embedding layer in the neural network, which allows learning the word representations at the same time as the model is trained. The second approach, and the most popular in NER, uses representations of words previously trained in other corpora instead of starting from scratch. This allows the transfer of knowledge between different tasks but belonging to similar domains.

In our experiments, we used domain-specific embeddings previously trained with 11 million unstructured free text diagnostics obtained from the Chilean Waiting List. This corpus was composed of 56,079,828 tokens, where the vocabulary length was 252 thousand different words. The original Mikolov's implementation of the Word2Vec algorithm was used to compute the embeddings with the default hyperparameters, except for the vector size, which was changed to 300. These 300-dimensional clinical embeddings can be downloaded from here[4]. Furthermore, during the training stage of our models, these embeddings were not left static, so the weights were updated.

Character Embeddings

In contrast to the previous method, the character-level model encodes each character in a sentence with a numerical vector. These embeddings have proven to be particularly useful for corpora with a large number of out-of-vocabulary words, misspelled words, emoticons, new words, and infrequent words [130]. Given the unstructured nature of medical diagnoses, this type of data is very common in our corpus.

The addition of these representations has improved the performance of models in a wide variety of NLP tasks. Two main architectures are used to create these embeddings: Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM). The former uses a one-dimensional CNN to find the numerical representation of words by looking at their character-level compositions, while the second approach uses the concatenation of its forward and backward representations retrieved from a bidirectional LSTM. In this chapter, we

[4] `http://doi.org/10.5281/zenodo.3924799`

explore the CNN-based approach for our experiments, while in the following chapters, we will use the LSTM-based approach.

3.2.2 Encoder Layer

After establishing the embedding layer, another fundamental step to consider is contextualization, which enriches the word representation by considering the dependencies of the words within the current sentence. This process is carried out in the encoding layer, for which we have chosen the following architectures:

Recurrent Neural Networks

Recurrent Neural Networks [100], also known as RNNs, are a class of neural networks that has significantly improved models' performance in sequence labeling tasks, such as NER. Unlike traditional neural networks, it can process sequential input data with variable lengths, such as text, video, and music.

The recurrence comes from the fact that each output is calculated based on the elements that precede it. For this reason, it uses a kind of memory to generate the desired output. The main advantage is that these neural networks can consider the correlation between the different data in the sequence, which is essential to improve word representations according to their context.

However, the main drawback of RNNs is the vanishing gradient problem, which hampers the learning of long data sequences. In other words, this phenomenon consists of the low weight given to the initial inputs in calculating the far outputs due to the activation functions applied in the intermediate states.

Long Short-Term Memory

Long Short-Term Memory (LSTM) [43] is a special kind of RNN explicitly designed to avoid the vanishing gradient problem. To alleviate this issue, LSTMs can eliminate or add the information they consider relevant to their processing sequence through additional cells, input and output gates.

In this research, we use the Bidirectional LSTM (BiLSTM), an architecture consisting of two LSTMs: one of the LSTMs takes the input in the forward direction, and the other in the backward direction, thus taking into account both contexts. Hence, the final representation of words depends not only on the previous words but also on the future words in the sentence.

Multi-Head Attention

In recent years, the emergence of attention-based architectures [118] has revolutionized the area of NLP. Attention is a technique that enhances the important parts of the input data and fades out the rest to capture long-term dependencies. This mechanism can be applied directly to the input or higher-level representation, such as embeddings or LSTM representations.

Multi-head attention is a module that allows the model to jointly attend to information from different representation subspaces at different positions, which would, otherwise, not be possible with a single attention head [118]. This mechanism runs through an attention module several times in parallel, avoiding the recursion characteristic of the LSTM. This is beneficial to reduce the training time considering the vast amount of data processed in NER datasets.

In the next chapter, we will see how this module has inspired the creation of some famous language models such as Flair and BERT. These models allow, among other things, to obtain contextualized embeddings, improving the representation of words according to their context.

3.2.3 Classification Layer

Finally, the word representations obtained in the Encoder layer has to be mapped into predefined categories. This process is carried out in the classification layer, for which we have chosen the following two methods:

Softmax

Mathematical function commonly used as the output layer in deep neural networks. As shown in Equation 3.1, this function transforms the vector representation of each word x_i into a vector of probabilities. Since it returns a probability distribution, the output values are in the range of $[0, 1]$, with the sum of the probabilities equaling 1. Intuitively, the output of this function represents the probability of belonging to each class, with the target class having the highest probability.

$$softmax(x_i) = \frac{exp(x_i)}{\sum_j exp(x_j)} \tag{3.1}$$

Although many NER systems use the Sigmoid function in their output layer, it is not the best approach for two main reasons. First, this function calculates the output probability distribution for each label based on the features of that particular word, i.e., it does not consider the information of its neighboring words. Second, the main assumption is that the true class labels are independent, therefore it can predict invalid transitions according to the IOB2 format. For example, it may tag a token at the beginning of an entity with the *I-* prefix.

Conditional Random Field

Conditional Random Field (CRF) is a probabilistic classification method used in NLP to obtain the most likely label sequence associated with a sentence, which is precisely the goal of NER. This model uses the contextual information from previous labels, thus increasing the amount of information to predict the labels accurately.

In order to map the representation of each word to the respective categories, the CRF algorithm needs two elements. First, it uses a transition matrix, where each cell represents the probabilities of transitioning from one label to another. Second, it uses the Viterbi algorithm [119], which takes the output vector obtained in the encoder layer, and the values in the transition matrix to obtain the best label sequence of the sentence.

The main advantage of using this approach is that, when classifying a token into one of the possible categories, it considers strong label dependencies by adding transition scores between neighboring labels. This allows us to handle the Softmax issue, avoiding specific invalid transitions in the IOB2 format.

3.2.4 Experimental Settings

Baseline

To study the contribution of each architecture described in the previous section, we start by designing a baseline model from which different changes will be made to improve its performance in the flat NER task. This baseline consists of four main components (1) an input layer that receives the tokens represented as indexes in the corpus vocabulary, (2) a word embedding layer to represent these indexes into numerical vectors from scratch, (3) an RNN encoding layer to contextualize the previous representation of tokens according to their context in the sentence, (4) a Softmax function to decode the most likely label sequence.

Ablation Study

Following the objectives stated at the beginning of this thesis, we would like to determine the best architectures for the embedding, encoding, and output layers. For this purpose, six modifications to the baseline were tested, as shown in Table 3.4. In these experiments, we compared the performance of models when using traditional RNNs against LSTMs, LSTMs against BiLSTM, word embeddings trained from scratch against pre-trained word embeddings, and Softmax against CRF algorithm. In addition to these direct comparisons, we measured the impact of adding elements such as character-level embeddings (setting 4) and the attention mechanism (setting 6).

	Embedding layer	Encoder layer	Classification layer
Baseline	Word Embeddings	RNN	Softmax
Setting 1	Word Embeddings	LSTM	Softmax
Setting 2	Word Embeddings	BiLSTM	Softmax
Setting 3	Medical Embeddings	BiLSTM	Softmax
Setting 4	Medical Embeddings + Character Embeddings	BiLSTM	Softmax
Setting 5	Medical Embeddings + Character Embeddings	BiLSTM	CRF
Setting 6	Medical Embeddings + Character Embeddings	BiLSTM + Attention	CRF

Table 3.4: Settings used in our experiments. The first model corresponds to the baseline.

Hyperparameters

Since we are working with neural networks, we must define the hyperparameter space and best values found in our experiments. For this purpose, we performed the random search strategy, which selects the best values by exhaustively testing different combinations of hyperparameters over a range of values.

In Table 3.5, we list the hyperparameters used as well as the range of values to perform the random search. To establish which is the best combination, we measured the performance using the validation partition. The initial weights of our models were set from a normal probability distribution with zero mean and variance of 0.1. In addition, we added seeds to ensure the reproducibility of the experiments.

Parameter	Range
batch size	{8, 16, 32, 64}
epochs	{10, 50, 100}
optimizer	{SGD, Adam, AdamW}
learning rate	{0.0001, 0.001, 0.1}
static embeddings	{True, False}
char emb dim	{20, 30, 40, 50}
LSTM depth	{1, 2, 3}
LSTM hidden size	{64, 128, 256}
attention heads	{8, 16}
dropout	{0.2, 0.3, 0.4, 0.5, 0.6}

Table 3.5: Hyperparameter search space.

Evaluation Metric

The performance was evaluated using the strict evaluation metric explained in Section 2.2, this is, calculating precision, recall, and micro F1-score. Due to the randomness present in these experiments, the models were run ten times with different initialization parameters. The reported results correspond to the mean and standard deviation (SD) of the evaluation rounds.

3.3 Results on flat NER

Table 3.6 shows the overall results obtained in our experiments. Interestingly, the best performing model (highlighted in bold) uses the concatenation of medical and character-level embeddings for word representation, a BiLSTM for contextualization, and the CRF algorithm for decoding. This setting achieves a mean micro F1-score of 74.8, far superior to the baseline score, which was 53.1.

Model	Precision	Recall	F1-score
Word Embeddings + RNN (Baseline)	59.3 (2.47)	48.3 (1.58)	53.1 (0.43)
Word Embeddings + LSTM (Setting 1)	68.3 (0.97)	65.9 (0.54)	67.0 (0.55)
Word Embeddings + BiLSTM (Setting 2)	72.6 (0.57)	69.8 (0.72)	71.1 (0.51)
Medical Embeddings + BiLSTM (Setting 3)	74.5 (0.75)	73.4 (0.79)	73.9 (0.62)
Medical Embeddings + Character Embeddings + BiLSTM (Setting 4)	74.2 (0.72)	73.8 (0.52)	74.0 (0.61)
Medical Embeddings + Character Embeddings + BiLSTM + CRF (Setting 5)	**75.1 (0.52)**	**74.4 (0.55)**	**74.8 (0.48)**
Medical Embeddings + Character Embeddings + BiLSTM + Attention Layer + CRF (Setting 6)	73.7 (0.74)	73.2 (0.62)	73.5 (0.65)

Table 3.6: Results for flat NER experiments on the Chilean Waiting List corpus. Data shown are mean (SD).

Starting the analysis from the baseline model, we can see that replacing the RNN architecture with an LSTM contributes to a significant increase according to the micro F1-score, which is further improved by incorporating bidirectionality into the LSTM. This finding confirms the importance of considering the past and future context of the words in the sentence since it leverages the representation of words.

Another significant increase is due to the addition of pre-trained word embeddings in the medical context. This was expected due to the nature of these vector representations, which come from a clinical context similar to our corpus, providing a better representation compared to the approach where embedding are trained from scratch.

Concerning the impact of adding character-level embeddings, the increase is not much significant than the previous setting but still achieves better results. One possible reason is that there are many misspelled and out-of-vocabulary words in our corpus, which is the main advantage of using these representations.

In addition, we can observe that adding the CRF algorithm to the classification layer contributed to achieving the best result according to the F1 measurement. We suspect that the main reason is that this algorithm allows finding the dependencies between possible labels in a sentence and does not allow invalid transitions in the IOB2 format, which could affect the model's performance.

Finally, and contrary to expectations, we can see that including the multi-head attention layer caused a decrease in the performance of our architecture, which can be explained by the overfitting generated by having such a complex architecture for a problem where the amount of data is not very large.

3.4 Discussion

Although our experimental results show a good performance according to the F1-score metric, we believe that they can be improved considerably. One of the main problems affecting the performance of flat NER models was the amount of data available for training. By ignoring the occurrence of nested entities, we lose many entities to train the model.

Metric	Train	Dev	Test
Sentences	8,014	890	990
Entities	35,480	3,971	4,289
Tokens	149,574	16,754	18,436
Annotated tokens	92,870	10,268	11,672
Vocabulary	17,421	4,281	4,680
Lexical diversity	11.6%	25.6%	25.4%
Mean tokens per sentence	18.7	18.8	18.6
Mean tokens per entity	2.62	2.59	2.72
Mean entities per sentence	4.43	4.46	4.33

Table 3.7: Statistics of the Chilean Waiting List corpus considering nested entities.

To illustrate this problem, Table 3.7 presents the statistics of the corpus considering nested entities. Compared to Table 3.3, we can see that the total entities per partition is much higher, which supports our theory that many inner entities are lost when transforming from standoff files to the CoNLL flat NER format. This is also evident in the average number of entities per sentence, which increases by approximately 30% for each partition when considering the deleted entities.

Another way to visualize this fact is to analyze the number of examples per entity type. Figure 3.9 shows the frequency of entities before transforming the problem to the flat NER task. Compared to Figure 3.6, we notice that some entity types such as abbreviations and body parts have lost almost 50% of the examples. In contrast, other categories such as diseases and clinical findings were not significantly affected. This is because the average token length of these entities is much longer than the rest, which means that they tend to be retained when eliminating shorter entities.

Figure 3.10 illustrates the appearance of nested entities in our corpus, with numbers indicating how many times the entity in the row is nested in the entity in the column. Please note that this matrix is not symmetric, as it is much more common to find, for example, a body part in a finding than the other way around. In fact, body parts are nested 3,136 times in findings, while findings are four times part of a body part. When nested annotations have the same length, we count them as nested into each other for both entities. An example of that is *HTA* (hypertension), which is both a Disease and an Abbreviation. In summary, the maximum nesting depth is three and 48.17% of the entities in the corpus contain other entities or are contained within another entity. This finding suggests that the Chilean Waiting List corpus is an excellent resource for the nested NER task.

From this analysis, we consider it necessary to design architectures that can deal with the nesting problem in our corpus. Therefore, in the following chapter, we propose two simple

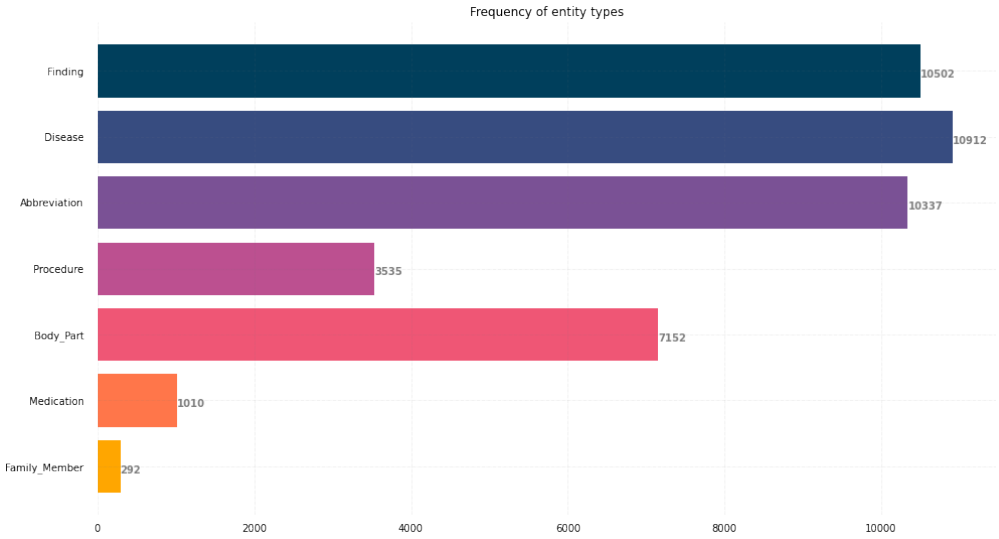


Figure 3.9: Frequency of entity types considering nested entities.

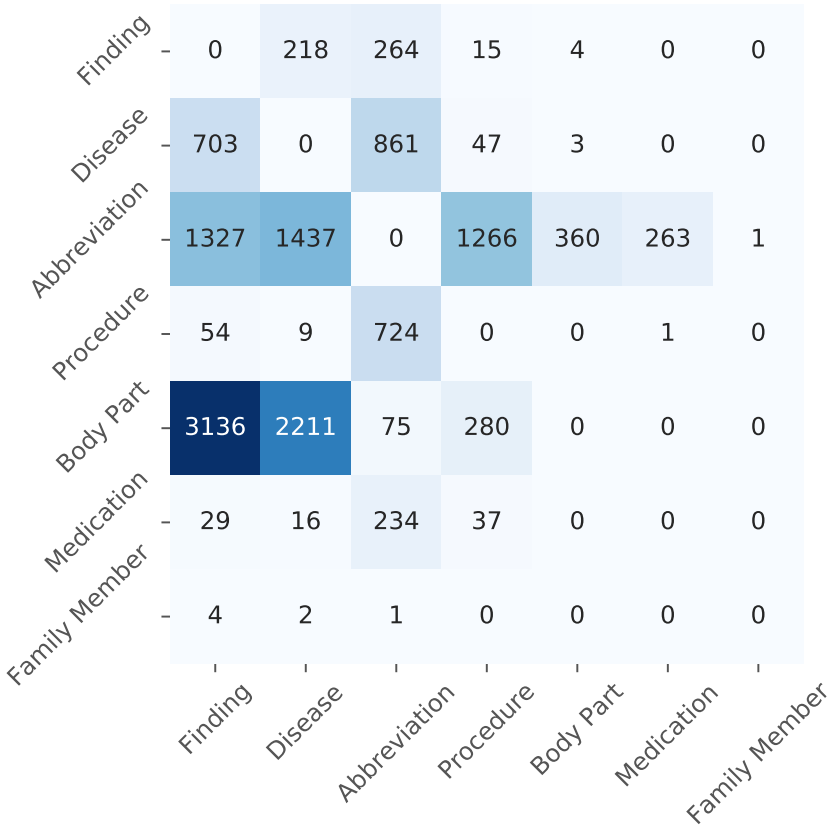


Figure 3.10: Characterization of nested entities. The numbers in each cell indicate how many times the entity in the row is nested in the entity in the column.

neural networks inspired by the sequence labeling approach, which are explicitly designed to deal with nested entities. In addition, we will describe an existing web page in which the best-performing model was incorporated. This application allows healthcare professionals to

use this tool to recognize key information in clinical diagnoses automatically. ¡

Chapter 4

Nested Named Entity Recognition in the Chilean Waiting List

This chapter presents two sequence labeling-based architectures for the nested NER task. Both models are capable of recognizing nested entities without relying on complex structures or heavy feature engineering. The aim is to improve the results obtained in Chapter 3, where nested entities were not considered. For this purpose, we first develop two simple but overlooked models that are potentially useful for solving the task. Then, to measure the effectiveness of the proposed methods, we compare the results against the Layered architecture, one of the state-of-the-art models in nested NER. Finally, we describe the main applications in which the best-performing architecture is currently used.

4.1 Nested NER Architectures

With recent advances in deep learning, neural networks based on the sequence labeling approach have substantially improved the results on the flat NER task. However, we argue that the NLP research community has little explored adapting these models for the nested NER task. Inspired by this approach, this section describes two methods for recognizing nested entities in our corpus.

4.1.1 Multiple LSTM-CRF (MLC)

The first architecture proposed consists of training multiple flat NER models, one for each entity type. The predicted labels of the input sentence correspond to the union of the outputs of each of these models, thus retrieving the nested entities. The main advantage of using this approach is that it can easily incorporate all the advances made for flat NER into the nested NER task. Another advantage is that each independent model can be trained in parallel to reduce the computational time of the training process.

Figure 4.1 shows an overview of the MLC model. Specifically, to create each flat NER

module, we follow the LSTM-CRF approach proposed by [61], one of the most widely used architectures for sequence labeling. To encode sentences, we use different combinations of embeddings in the stacked embedding layer. First, we concatenate domain-specific word embeddings with character embeddings retrieved from a bidirectional character-level LSTM. Next, we enrich word representations by adding contextualized embeddings from Flair [3] and BERT [28], which have proven to be particularly effective on NER. The output is fed into a BiLSTM encoding layer to obtain long-contextual information. Finally, we use a CRF-loss and the Viterbi algorithm to decode the most likely label sequence using the IOB2 tagging format in the classification layer.

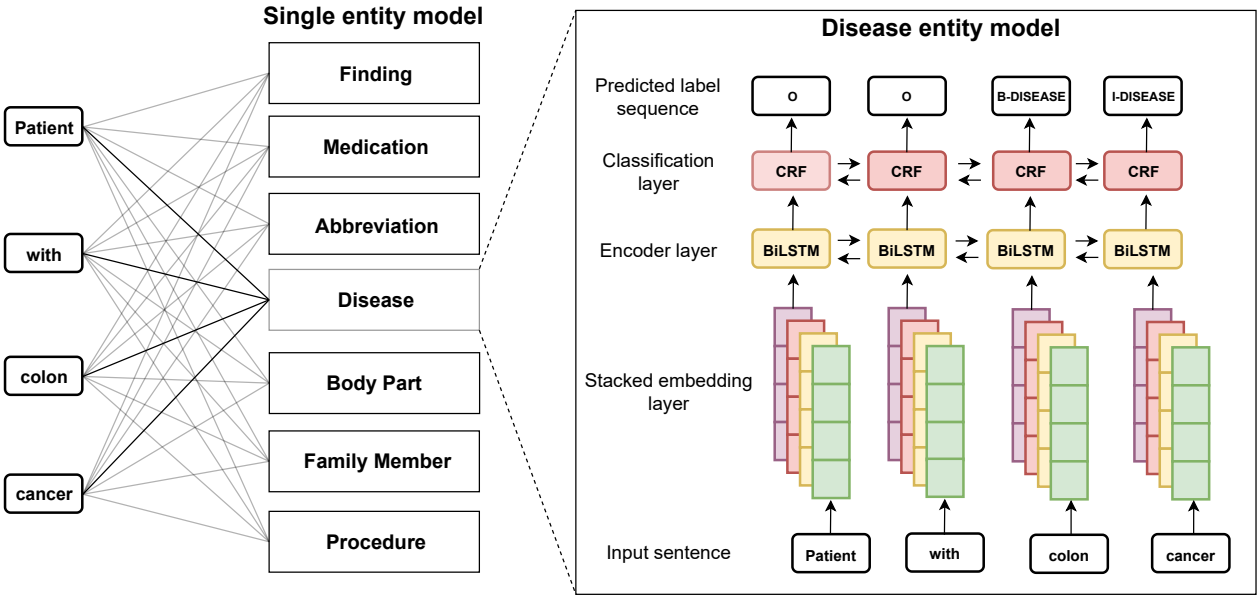


Figure 4.1: Overview of the MLC architecture, where each entity type has an associated flat NER model. The right side of the figure shows, as an example, the flat NER module for the Disease label in the Chilean Waiting List corpus.

4.1.2 Sequence Multi-Labeling (SML)

The second approach formulates the nested NER task as a multi-label token classification problem. The name comes from the idea of taking full advantage of the token-level representation provided by the embedding and encoder layers to perform a token-level classification. It is multi-label since each token can be tagged with more than one label. Despite its simplicity, no one has proposed to handle nested entities using this approach to the best of our knowledge.

SML is similar to the MLC architecture, but the aim is to use a single model for all entity types, thus improving the computational time of the training process. As shown in Figure 4.2, to keep the model as simple as possible, we do not use pre-trained language models but only word-level and character-level embeddings. We use the BiLSTM encoder layer to capture the dependencies of words, and the output is fed to a Feed-Forward Neural Network (FFNN) layer to reduce the size of word representation to the number of entity types. Next, we employ the following method to address the classification problem: First, we use a Sigmoid

function to estimate the probability that a token belongs to each class independently of the others. This mechanism allows us to have more than one label per token. Second, to improve model performance, the overall Sigmoid threshold of each class is adjusted in the validation set using a random search. Finally, we use a binary cross-entropy function to compute the loss.

It is worth mentioning that having a multi-label token classification makes it hard to apply the CRF algorithm in this architecture since the CRF algorithm calculates the most likely label sequence given a sentence. Therefore, the assumption is that each token can have at most one possible label, which contradicts our multi-label approach. However, one of the baselines that we will see in the next chapter makes modifications to the CRF algorithm to incorporate it in an architecture similar to ours.

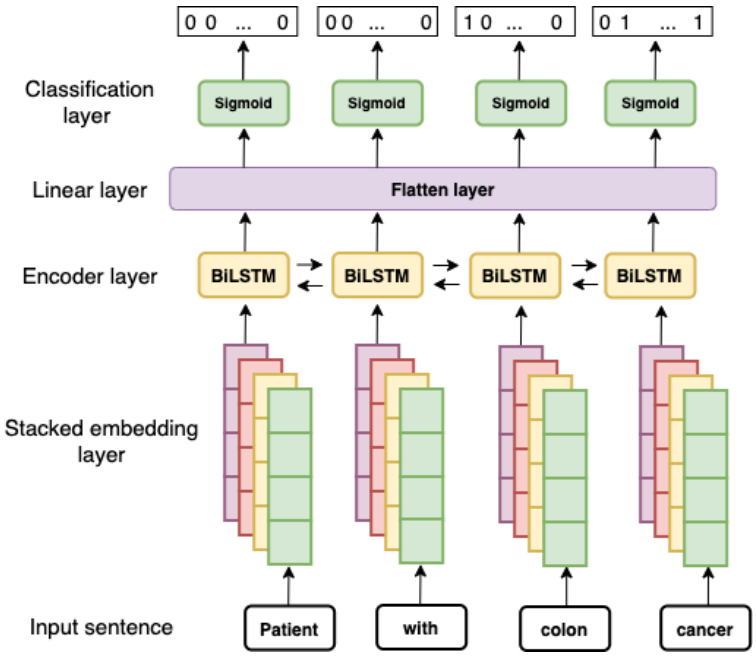


Figure 4.2: Overview of the SML architecture. The numbers at the end of the figure mean that the token belongs to each category (1) or not (0).

4.2 Methods

In this section, we present the baseline, the experimental settings, and the methodology used to validate the effectiveness of the best-performing model.

4.2.1 Baseline

To adequately measure the performance of the proposed models, it is necessary to perform a comparison against state-of-the-art models in the nested NER task. For this reason, we chose the Neural Layered architecture proposed by Ju et al. [49], one of the most popular

models in this task. Both this architecture and our methods belong to the sequence labeling category.

The decision was based on how this architecture treats nested entities and the ease of adapting the code to our corpus. Furthermore, 10.75% of the entities are involved in spans of text tagged with multiple entity types, which is a problem little addressed in the literature, and this approach can deal with it. In addition, the Neural Layered model is inspired by the LSTM-CRF architecture [61], which facilitates the comparison of hyperparameters with our proposed models since they share similar components.

As detailed in Figure 4.3, this model works by dynamically stacking flat NER layers to predict entities in an inside-to-outside way until no entities are extracted. Each layer is built with the popular LSTM-CRF approach, an architecture that is precisely the backbone of our MLC model. Specifically, it merges the output of the LSTM in the current flat NER layer to build a new representation for detected entities and subsequently feeds them into the next layer. This process allows the model to identify external entities by taking full advantage of the inner entity's representations.

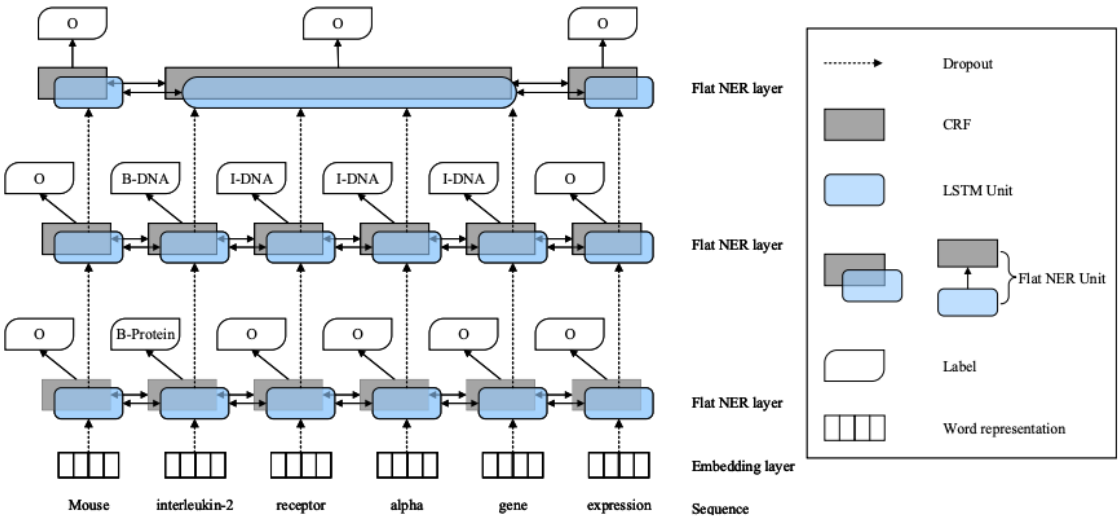


Figure 4.3: Overview of the Layered model.

The source code is freely available[1] to reproduce the experiments, and input files can be obtained using our preprocessing module described in Section 3.1.1.

4.2.2 Word Representation

To encode the sentences, we used the medical word embeddings described in Section 3.4. These representations were not left static during the training process, and out-of-vocabulary words were initialized using a zero vector. Additionally, we concatenated a character-level representation, following the LSTM-based method proposed by Lample et al. [61].

[1] `https://github.com/meizhiju/layered-bilstm-crf`

One of the main drawbacks of traditional word embeddings is that words can have different meanings in different contexts. For example, the word *Bank* does not have the same meaning in the sentence *Central Bank of Chile* as it does in *Blood Bank*. This suggests that it is not optimal to have a unique representation for each word, as we have done so far. To address this issue, the so-called *contextualized embeddings* were introduced, which have improved the performance of several NER systems. In the following lines, we describe two linguistic models used to retrieve these representations: Flair and BERT.

Flair

Flair [3] is a character-level language model, which represents words as sequences of characters contextualized by the surrounded text. As shown in Figure 4.4, to create these embeddings, they retrieve the internal states of a bidirectional character-level LSTM for each word. Specifically, the model extracts the hidden state output after the last character in the word and the hidden state output before the first character in the word. This process allows us to obtain the context of the word in the sentence in both directions.

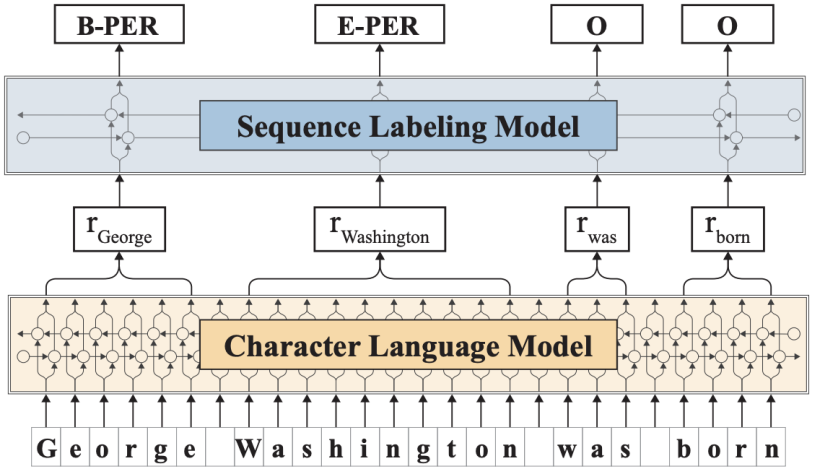


Figure 4.4: Overview of the Flair character-level language model.

Since there was not an available version of this model for clinical text in Spanish, we trained and added new language models[2] to the Flair framework: *es-clinical-forward* and *es-clinical-backward*. To train these models, we used the same medical corpus on which the pre-trained word embeddings were trained. In addition, we followed the same settings and assumptions as stated in the Flair article.

BERT

BERT [28] is a transformer-based architecture, which represents a general language model that supports transfer learning and fine-tuning on specific tasks. The use of this model for creating contextual word embeddings has led to significant improvements on several NLP tasks,

[2] `https://github.com/plncmm/bio-flair`

including NER. In our experiments, we used the cased version of Spanish BERT (BETO) [17] without fine-tuning. This model was trained on an extensive Spanish unannotated corpus composed of 3 billion words. To create each word representation, we concatenate the values of the last four hidden layers of the model without fine-tuning. Besides, since BERT uses word-piece tokenization, we computed the word embeddings using the average representation of the subtokens embeddings.

4.2.3 Settings

To choose the best hyperparameters for each model, we used the same methodology as the previous chapter. That is, we performed a random search over the hyperparameter space described in Table 4.1. In addition, to ensure a fair comparison between our methods and the baseline, we tried to use hyperparameters as similar as possible.

The MLC architecture was trained up to a maximum of 100 epochs using the CRF loss. For optimization, we used SGD with a mini-batch size of 16 and an initial learning rate of 0.1. In the case of SML, the loss function was the binary cross-entropy, and the training consisted of 20 epochs. The optimizer used to train SML was Adam, with a learning rate of 0.01 and mini-batches of 16. For both models, we used a learning rate scheduler and an early stopping strategy based on the performance of the development partition to avoid overfitting. We reduced the learning rate by 0.3 if there was no performance improvement after three epochs. We also applied dropout regularization [108] after the embedding layer and BiLSTM. The BiLSTM settings were the same, using three layers with 128 units each.

Parameter	Range	SML	MLC
max epochs	[20, 100]	20	100
optimizer	{SGD, Adam, AdamW}	Adam	SGD
batch size	{8, 16, 32}	16	16
learning rate	{0.0001, 0.001, 0.1}	0.001	0.1
char emb dim	[20, 50]	50	25
dropout	[0.2, 0.8]	0.5	0.3
BiLSTM depth	{1, 2, 3}	3	3
BiLSTM hidden size	{128, 256, 512}	128	128

Table 4.1: Hyperparameter search space and the best values found for our models. In the case of continuous intervals, 5 values were selected in the interval with the same distance.

The MLC model was implemented using the Flair framework [4], and the SML model by using the PyTorch library [96]. All the experiments were performed using a Tesla V100 GPU, and RAM with 192GB of capacity. To ensure reproducibility, the source code of our experiments is freely available in our repository[3].

[3] `https://github.com/plncmm/acm_health_msen`

4.2.4 Model Evaluation

To compare the predictions against the real data, we used the standard evaluation metric described in Section 2.2.1. This is, calculating the precision, recall, and micro F1 score over all entities in the test partition. An entity is considered correct when both entity types and boundaries are correctly predicted.

However, we argue that reporting a single performance score is insufficient to compare non-deterministic approaches since results might change when using different subsets. Therefore, we would like to determine whether the differences between the performance of the best model and the baseline are reliable or are just due to statistical chance.

According to work described in Dietterich [30], creating a statistical test to compare two machine learning models is beneficial to guarantee reproducibility. Therefore, we performed a *k-fold cross-validated paired t-test*, comparing the model with the best performance in the test partition with the Layered baseline. This test consists of the following procedure:

1. We perform the standard *k-fold cross-validation* technique to estimate the skill of models on unseen data. First, we randomly separate the original data into k mutually exclusive subsets, known as *folds*. Then, we repeat the following algorithm. First, we select one of the subsets for testing and the remaining $(k-1)$ subsets for training and validation. Second, we train the MLC and Layered models on these partitions, computing the difference in the performance of the models. We repeat this process k times so that the test sets do not overlap each other. Thus, we will have k differences calculated (diff).

2. Then, after demonstrating that both models' results follow the Gaussian distribution, we define our null hypothesis as follows: *there is no difference between the performance of both ML models*. To validate or refute this hypothesis, we calculate the t-score as shown in equations (4.1, 4.2). If the p-value associated with this t-score is less than the significance level (typically 0.05), we reject the null hypothesis, suggesting that both ML models perform differently.

$$\text{std} = \sqrt{\frac{1}{k-1}\sum_{i=1}^{k}(\text{diff}_i - \overline{\text{diff}})^2} \tag{4.1}$$

$$\text{t}_{score} = \frac{\overline{\text{diff}}}{std}\sqrt{k} \tag{4.2}$$

The problem is that this statistical hypothesis test also assumes the independence of experiments. However, in cross-validation, the training sets overlap between different folds. The main consequence of violating this assumption is a slightly high type I error. Thus, we implemented the corrected version of this test proposed in Kononenko and Kukar [54], which showed that the violation of the independence t-test might lead to underestimating the variance of the differences. To solve this problem with the paired Student's t-test, they proposed to correct the variance estimate by taking this dependency into account. Specifically, the

factor k in Equation (4.2) is replaced with the reciprocal squared root of its inverse plus the ratio between numbers of testing subsets (n_1) and training (n_2) subsets for each step in cross-validation, leaving the final formula as shown in Equation (4.3).

$$\mathrm{t}_{score} = \frac{\overline{\mathrm{diff}}}{std\sqrt{(\frac{1}{k} + \frac{n_1}{n_2})}} \tag{4.3}$$

Finally, since neural network models are stochastic processes, it is worth mentioning that replicating these experiments may lead to slightly different results in different runs. To ensure the reproducibility of our experiments, we made public in the repository the partitions used for this process and the original subsets on which the experiments were tested.

4.2.5 Error Analysis

For better understanding and explainability of the best model, we propose to do an error analysis using the work proposed by Nejadgholi et al. [92] but modified for nested entities. An error analysis is necessary to understand the output of neural models, which operate as a black box. The output of an entity recognition model may be incorrect because either the span is incorrect or the label is incorrect (or both). Based on these principles, we distinguish five types of errors listed below and exemplified in Figure 4.5.

1. False-positive: the model predicts one or more entities not annotated in the test subset.
2. False-negative: the model predicts no entities for a given span, but the test subset contains entities. This malformed addition can be complete (the model predicted no entities for the span) or partial (the model predicted an incomplete list of entities for the given span)
3. Wrong label, right span: an annotated entity in the test subset and the predicted entity have the exact spans but different entity types.
4. Wrong label, overlapping span: the annotated entity in the test subset and the predicted entity have overlapping spans and different entity types.
5. Right label, overlapping span: the annotated entity in the test subset and the predicted entity have the same entity types but overlapping spans.

4.3 Results on nested NER

4.3.1 Main Results

Table 4.2 shows the overall results of our experiments. Interestingly, each configuration of the MLC architecture outperforms the Layered baseline by a wide margin according to the F1

True annotation	abdominal pain of +- 8 months given by right flank abdominal pain with usg .... showing left kidney stone
Error type	Predicted annotation
False-positive	abdominal pain of +- 8 months given by right flank abdominal pain with usg .... showing left kidney stone
False-negative	abdominal pain of +- 8 months given by right flank abdominal pain with usg .... showing left kidney stone
Wrong label, right span	abdominal pain of +- 8 months given by right flank abdominal pain with usg .... showing left kidney stone
Wrong label, overlapping span	abdominal pain of +- 8 months given by right flank abdominal pain with usg .... showing left kidney stone
Right label, overlapping span	abdominal pain of +- 8 months given by right flank abdominal pain with usg .... showing left kidney stone

Figure 4.5: Example annotations for each error type. A correctly annotated span of text is described in the head, and malformed annotations are described below. For illustrative purposes, we are only showing annotations for Finding (in light purple) and Procedure (in dark green). Malformed annotations are shown in bold. Note that we are using the first referral shown in Figure 3.3.

measure. These results are further improved by adding new representations to the embedding layer. The model with the best performance (highlighted in bold) is the MLC setting that used medical word embeddings concatenated with character and Flair embeddings, achieving a micro F1-score of 80.27. In contrast, although the SML model does not perform better than MLC, it obtained competitive results and outperformed the baseline used. Since these results were obtained on a corpus with a high percentage of nested entities, we believe that both proposed approaches are reliable models for the nested NER task, despite their apparent simplicity.

Regarding the best model, Table 4.3 shows precision, recall, and micro F1-score per entity type, as well as the number of examples in the test partition. The entity type with the best results was Abbreviation, which is expected since it is easy to recognize from the morphological point of view. This entity is usually one token long; therefore, the chances of being mistaken due to wrong boundaries are low. The opposite occurs with the entity type Finding, which is four tokens long on average, thus very easy to have it wrong in the limits. Moreover, Findings are the hardest to have consistently annotated by humans.

The most clinically relevant category is Disease, which reached a micro F1-score of 82.92. Although these results can be improved, we believe they are good for two reasons: First, it is difficult to recognize diseases even by medical specialists. Second, considering the high average number of tokens per entity, using strict metrics is challenging to obtain good re-

Model	Precision	Recall	F1-score
Neural Layered Model [49] (baseline)	77.0	72.12	74.48
SML	76.6	72.7	74.60
MLC [Word]	76.59	74.84	75.71
MLC [Word+Char]	77.75	78.29	78.02
MLC [Word+Char+BERT]	79.72	78.83	79.27
MLC [Word+Char+Flair]	**80.24**	**80.30**	**80.27**
MLC [Word+Char+Flair+BERT]	79.90	78.13	79.01

Table 4.2: Results obtained with different models and settings on the Chilean Waiting List corpus. Here, Word stands for word embedding, Char is character embedding, and the Flair and BERT models were implemented as described in the text.

Entity	Precision	Recall	F1-score	Support
Abbreviations	93.65	95.07	**94.35**	993
Disease	82.65	83.19	82.92	1,071
Medication	87.21	81.52	84.27	92
Finding	62.31	62.13	62.22	1,059
Body Part	85.91	87.01	86.46	708
Family Member	96.55	87.50	91.80	32
Procedure	72.96	69.46	71.17	334

Table 4.3: Results for each entity type using the best MLC setting in the test subset.

sults. Finally, we observe that Family Member was easy to recognize by the model, which is explained by the fact that it is a kind of dictionary of terms, where few words can refer to this entity.

4.3.2 Hypothesis Test Results

We used the best MLC setting to perform a statistical comparison with the baseline. The cross-validation process demonstrated the efficacy and high level of generalization of the MLC model on unseen data, significantly outperforming the baseline in all measurements (Table 4.4), consistent with the results in Table 5.5. In practical terms, the statistical results and the k-fold cross-validation provide convincing evidence that the MLC and Layered models perform differently.

4.3.3 Error Analysis Results

Our best MLC model made 1,302 errors on the test subset. The highest proportion of errors corresponds to *right label, overlapping span* (38%), followed by false negatives (29.6%) and false positives (22%) (Figure 4.6a). Finding is the entity type with the highest proportion of

	Neural Layered Model [49] (baseline)	MLC [Word + Char + Flair]	P value
Mean	73.20	79.81	$8.8e^{-9}$
SD	0.752	0.469	
Min	72.16	79.16	
Max	74.65	80.66	

Table 4.4: Results of the 10-fold cross-validation on the best MLC setting and the baseline. Results are calculated based on the micro F1-score.

these three types of errors, covering almost 60% of *right label, overlapping span* error, 40% of false negatives, and 35% of false positives (Figure 4.6b). It has previously been reported that better NER models generate more *right label, overlapping span* errors, suggesting that it could be because the span information may be vaguer in the representation resulting from contextualized embeddings by combining the meaning of words through an attention mechanism. Consequently, proper treatment of this type of error is essential in the comparison of modern NER systems [92].

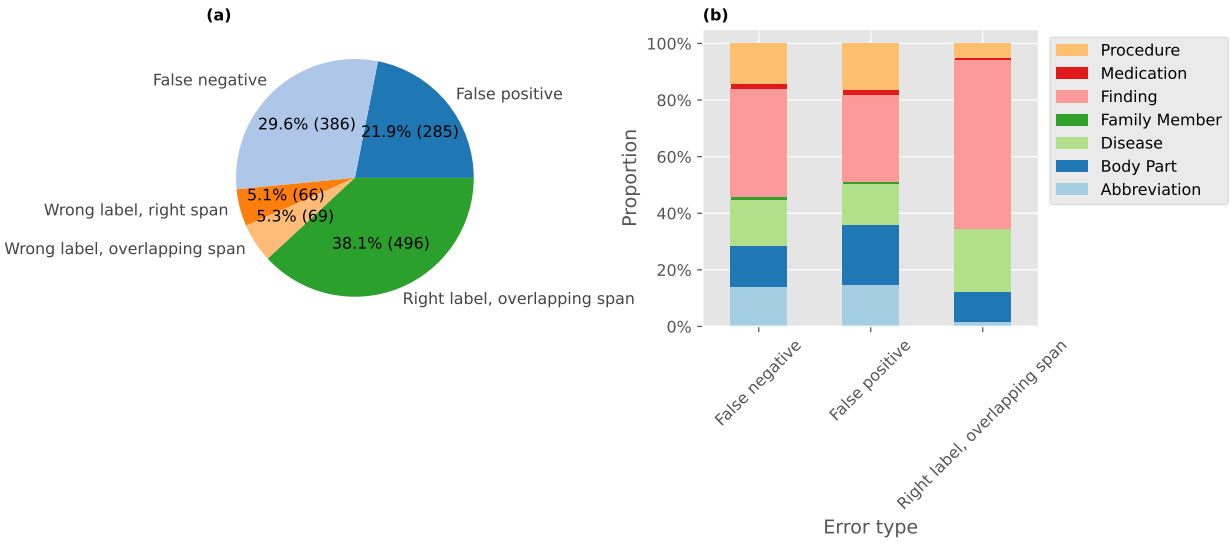


Figure 4.6: Distribution of the errors types found by the error analysis. This analysis was done using the incorrect best models' predictions on the test subset. Panel (a) shows the overall distribution of the error types, and panel (b) shows the distribution of entities inside error types.

Regarding the *wrong label* errors, the confusion matrix (Figure 4.7) shows that the Finding and Disease entity types are more often confused by the model, and this could be mainly because of the close semantic relatedness of both entity types; these categories are often subject to discussion even by the expert annotators.

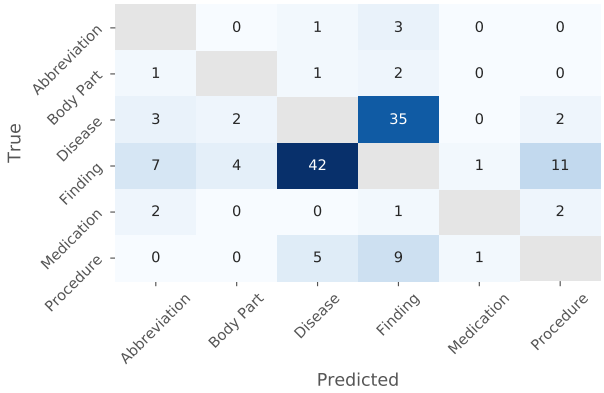


Figure 4.7: Confusion matrix for the wrong label errors found by the error analysis on the incorrect best models' predictions using the test subset.

4.4 Demo of our Medical Entity Recognition Model

One of the main motivations of our work is that people from the health area can test the nested NER system. Apart from publicizing our research, this provides a more reliable measure of model generalization on unseen data.

As shown in Figure 4.8, we integrated the MLC model into an existing web page[4]. The functioning of the web page is quite simple: Using an intuitive interface, people can manually enter some text in Spanish with medical relevance. By pressing a button, the tool will automatically tag the entities found using our MLC architecture. Finally, a google docs form provides the possibility to write some comments about the results obtained. This feedback could be an important support to improve the model performance in future work. The figure shows an example of a medical text labeled with the platform.

4.5 Discussion

In this chapter, we described two simple yet powerful architectures for solving the nested NER task, obtaining excellent results on the Chilean Waiting List corpus. The results obtained are much better than the previous chapter, demonstrating the importance of including nested entities in our experiments. Specifically, the best results were obtained using the MLC model, which proved to be superior to the baseline by a wide margin through statistical tests. In addition, although the SML architecture did not obtain the best performance, it has also proven to be a valuable approach to solving this task, and we believe it should be included in future work.

Given the promising results, we would be interested to know whether the MLC architecture is the most suitable approach for solving the nested NER in our corpus or there are models with better performance. Moreover, we wonder if this model can obtain good results

[4] `https://pln.cmm.uchile.cl/clinical-ner/index.xhtml`

Figure 4.8: Web application created to test our model.

on other nested NER corpora from different domains and languages, which might evidence a gap in the nested NER task by underestimating this model.

To answer these questions, in the next chapter, we provide an empirical study comparing our MLC architecture with several state-of-the-art models in nested NER. These architectures are tested in three datasets from different languages (including ours), with particular attention to the impact of using pre-trained language models. In addition, we propose new task-specific evaluation metrics to adequately measure the performance of models in nestings, which is the primary goal of the task. Conducting this study will allow us to understand better the nested NER problem we face and help us to decide which model better suits our problem.

Chapter 5

Nested Named Entity Recognition Revisited

This chapter aims to validate the effectiveness of the MLC model proposed in the previous chapter by comparing it with other state-of-the-art models and on other datasets from different languages. Moreover, to better understand the problem of nested entities in the Chilean Waiting List corpus, we identify some gaps in nested NER literature related to the task formalization, model selection, and evaluation metrics. To address these issues, we provide an empirical study of different nested NER architectures, proposing new task-specific evaluation metrics.

5.1 Motivation

Although the previous chapter showed promising results using the MLC architecture, analyzing other state-of-the-art models and other datasets is essential to validate the effectiveness of our model. For this reason, in this chapter, we study in-depth the current state-of-the-art solutions in nested NER. Even though these studies have shown competitive performance, we realized that most of them have three critical problems discussed below.

First, we noticed that most of the literature ignores the case in which the same text span is tagged with more than one entity type. This case is very common in the Chilean Waiting List corpus, and it was first noticed by Alex et al. [5], but was not analyzed further. One of the main advantages of our architecture is that it addresses this problem.

Second, with the incorporation of large pre-trained language models, the standard LSTM-CRF [61] sequence labeling architecture received substantial improvements for flat NER tasks [74]. However, little research has been conducted on adapting this architecture to the nested NER task using the single entity approach proposed in the previous chapter, i.e., training independent flat NER models for each entity type. This chapter studies the multiple LSTM-CRF (MLC) architecture in-depth, testing it on three nested NER corpora and comparing the performance with several state-of-the-art models.

The apparent simplicity of MLC would lead us to believe that it should be considered as a natural baseline for any proposed architecture in nested NER. However, we realized that the few research that has incorporated this model had used it as a baseline [91, 71, 32], but their reported results are not competitive. We believe that the problem lies in the fact that they do not use the full potential of recent advances in flat NER architectures, such as adding pre-trained language models. These elements are incorporated in our work to show the effectiveness of this model. Despite the apparent simplicity, we show that this architecture yields very positive results on several datasets, achieving state-of-the-art on our corpus and outperforming several recent approaches explicitly designed for nested entities.

Finally, we argue that the way the community is evaluating this task does not adequately measure the effectiveness of a model at identifying nested entities. Specifically, the current metric calculates the micro F1-score over all entities in the test partition, which is the same metric used in flat NER. Consequently, a model that performs well over flat entities but not nested ones may also obtain good results. To alleviate this problem, we identify the different types of nesting by formalizing the nested NER task and then propose new task-specific metrics for these cases.

Addressing these problems encountered in the nested NER literature allows us to understand the nesting problem better and validate the MLC architecture's effectiveness. In addition, it serves as a support to make the final decision on which model will be in production for the applications described in the previous chapter.

5.2 Datasets

To provide empirical evidence for the effectiveness of the proposed model, and since most previous work on nested NER focused on English datasets, we conducted our experiments using corpora from three different languages and domains. The statistics for each corpus are shown in Table 5.1, and below, we give a brief explanation of the two datasets studied apart from our corpus.

	GENIA			GermEval			Chilean Waiting List		
	Train	Test	Dev	Train	Test	Dev	Train	Test	Dev
tokens	454,882	57,021	48,932	452,853	96,499	41,653	149,574	18,436	16,754
sentences	15,023	1,854	1,669	24,000	5,100	2,200	8,014	990	890
avg sent len	30.3	30.8	29.3	18.9	18.9	18.9	18.7	18.6	18.8
entities	45,929	5,474	4,337	31,545	6,693	2,886	35,480	4,289	3,971
avg entity len	2.9	2.9	3.1	1.4	1.4	1.5	2.6	2.7	2.6
nested entities (%)	17.0	20.6	16.8	15.0	14.7	14.1	46.4	45.9	46.7
nested entities	7,795	1,130	727	4,721	986	407	16,456	1,969	1,856
- different type	3,712	589	369	4,230	892	366	12,635	1,555	1,398
- same type	4,132	547	358	536	93	44	0	0	0
- multi-label entities	0	0	0	2	2	0	4,241	470	502

Table 5.1: Statistics of the datasets involved in our study.

5.2.1 GENIA

English GENIA V3.02[1] [52] is an annotated biomedical corpus collected from 2,000 MEDLINE abstracts. This corpus was created to support the development and evaluation of information extraction and text mining systems for the domain of molecular biology. It comprises 36 fine-grained entity types and 55,740 entity mentions, of which 17.3% are involved in nesting. Figure 5.1 shows an example of an annotation with nested entities in GENIA.

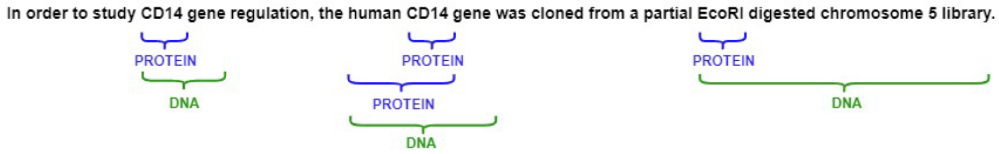


Figure 5.1: Example of nested entities in GENIA [1].

To pre-process the data, we followed the same setup as the previous work [33, 76, 131], collapsing sub-types into their five super-types, including DNA, RNA, protein, cell line, and cell type categories. We used the first 90% of the sentences for the training set and the remaining 10% in the test set for training NER models.

5.2.2 GermEval

The GermEval NER Shared Task is an event that makes available German data with NER annotations. The aim is to significantly advance state-of-the-art in German NER and push the field of NER towards nested representations of named entities. The competition has been organized annually, and the first edition of the competition (2014) was dedicated to the recognition of named entities.

In our experiments, we used the GermEval 2014 corpus [2], which is a nested NER resource sampled from German Wikipedia and online news. This dataset consists of 41,124 entity mentions, where 14.9% of them are involved in nesting. It contains two levels of nesting and 12 entity types. Figure 5.2 shows an example of an annotation with nested entities in GermEval.

Figure 5.2: Example of nested entities in GermEval.

For a fair comparison, in both the GENIA and GermEval datasets, we used the pre-processed version released in [131] while in our corpus, we used the public files released in this repository[3]. These files are already tokenized and follow a format similar to CoNLL but

[1] http://www.geniaproject.org/genia-corpus/pos-annotation
[2] https://sites.google.com/site/germeval2014ner/data [13]
[3] https://zenodo.org/record/3926705

with some modifications to support nested entities. All these details can be found in the documentation of the code of our experiments.

5.3 Methods

5.3.1 Baselines

We compared the MLC architecture with several state-of-the-art models in GENIA and GermEval datasets. According to the classification of nested NER approaches described in Chapter 2.3.3, we included one structure-based, two sequence labeling-based, and three region-based baselines. In addition, we used the Layered [49] baseline described in the previous chapter. Based on the published source code, in the following lines, we describe the models used as a reference for analyzing both traditional and task-specific metrics.

Exhaustive model

Exhaustive neural architecture proposed by Sohrab and Miwa [107], which considers all possible subsequences up to a defined length as potential named entity candidates. As shown in Figure 5.3, to enhance the word-level representation, they concatenate domain-specific word embeddings with character-level embeddings retrieved from a character-level BiLSTM. The output is fed to a BiLSTM to obtain a contextualized representation of these words according to the neighbor words in the sentence. Then, to obtain a span-level representation, they concatenate the representations of the start-end tokens in the span with the inside representation, which is the average of internal word embeddings. Finally, to classify these spans into their entity type, they use a Softmax output layer.

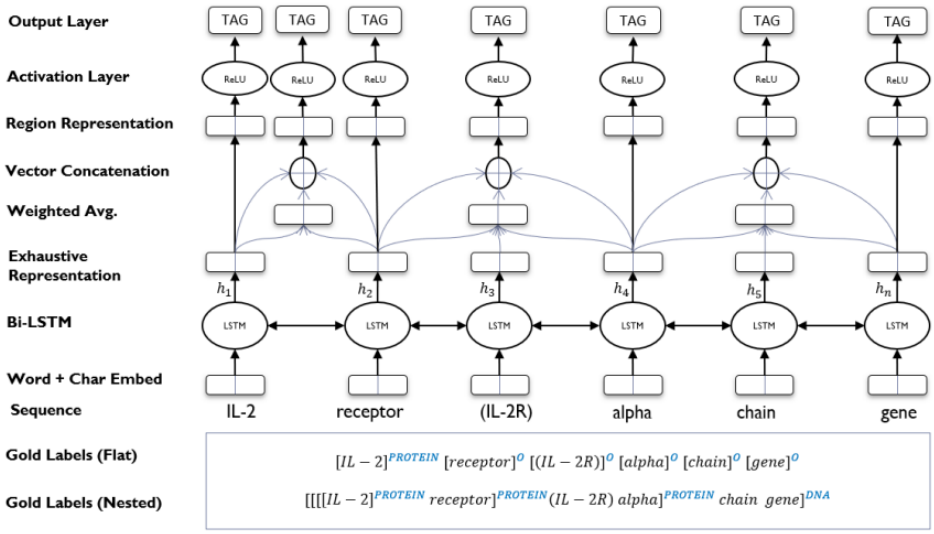


Figure 5.3: Overview of the Exhaustive architecture.

Boundary model

Model proposed by Zheng et al. [131] that combines ideas from Layered and Exhaustive architectures but correct their weaknesses. Figure 5.4 shows an overview of the model, which is called boundary-aware. Under this approach, the nested NER task is divided into two sub-tasks: first, the entity boundary detection and then the label prediction.

To represent each word in the sentence, they use the same method as the Exhaustive model, i.e., word-level and character-level embeddings are concatenated. Then, following the flat NER architecture of the baseline Layered, they use a BiLSTM sequence labeling layer to detect boundary-relevant regions within a limited length. The output of this layer follows the IOB2 format, i.e., a token is classified as B when it is the beginning of an entity, I when it belongs to the body of the entity, or O when it does not belong to any entity. Finally, they represent each pair of B and E tokens by averaging the representations of each token that falls within these boundary regions. This information is used to classify these regions into predefined categories using either a Softmax function or CRF algorithm.

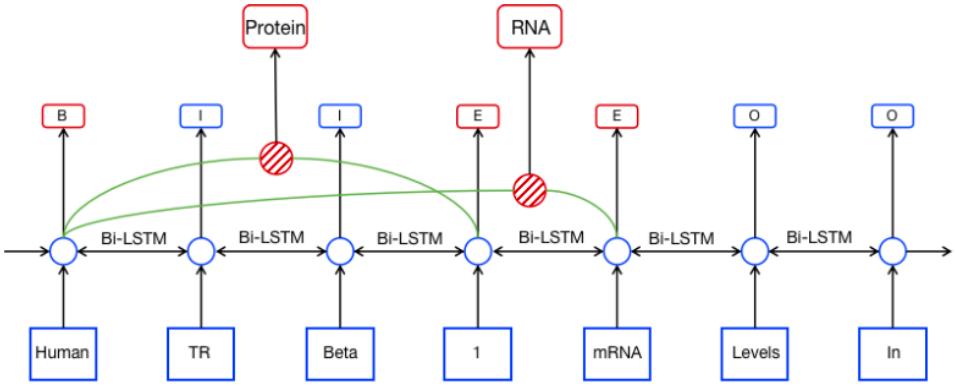


Figure 5.4: Overview of the Boundary architecture.

Recursive-CRF model

Sequence labeling-based approach that iteratively extracts nested entities from outermost to innermost using a CRF-based algorithm [106]. Figure 5.5 shows an overview of the model, which works as follows: First, they use a separate CRF for each entity type, which allows finding the best label sequence associated with that category in the sentence, thus retrieving the outermost entities. Then, they analyze each of these entities found to obtain the inner entities. Since each entity is a sequence of tokens, they calculate the second-best CRF score over that span using the previously calculated scores, obtaining the first level of inner entities.

This process is repeated until no more entities are extracted for that entity type, or all possible sub-sequences of the sentence are analyzed. Then, the algorithm is repeated for the entity types in the corpus to obtain all the entities. One of the main advantages of this architecture is that it can handle the situation where the same span is assigned to multiple entity types, which we have already seen is very common in our corpus. Moreover, it can recognize nested entities of the same type, a frequent case in the GENIA corpus.

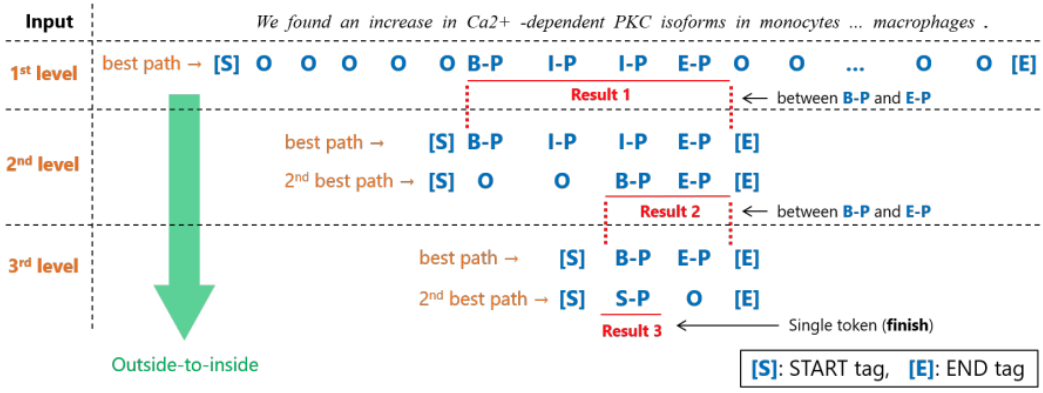


Figure 5.5: Overview of the Recursive-CRF architecture.

Pyramid model

Structure-based method proposed by Wang et al. [123]. Currently, this approach is the state-of-the-art in GENIA without using external supervision. Figure 5.6 shows that, unlike previous baselines, they incorporate contextualized embeddings to enrich the word representation in the encoding layer. These embeddings are retrieved using two language models, Flair, and BERT. Then, this information is passed to a decoding layer that recognizes entities in a bottom-up manner, assimilating the shape of a pyramid. The mode of operation is simple; they use L flat-NER layers, where the $i-th$ layer recognizes entities with lengths equal to i. Each layer is created using an LSTM to decode i-length entity mention and a CNN to pass the text region embeddings enriched with layer information to the next layer. The total number of entities found in the sentence is obtained by joining the output of these L layers.

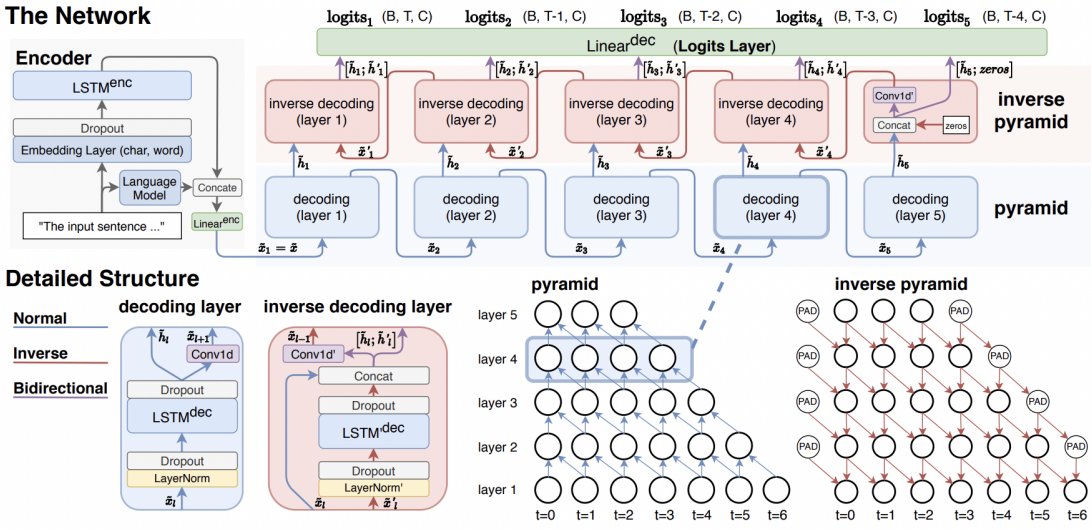


Figure 5.6: Overview of the Pyramid architecture.

Biaffine model

Region-based architecture proposed by Yu et al. [127], which uses a biaffine model to score pairs of start and end tokens in a sentence. Then, using specific constraints for nested entities, they classify these regions into the predefined list of categories.

Figure 5.7 shows a simple illustration of this architecture. To encode words, they concatenate domain-specific embeddings, character-level embeddings, and contextual embeddings retrieved from BERT. Unlike previous work, the contextual representation is created using the paragraph-level context of the document rather than sentence-level context. The output of the embedding layer is passed to a BiLSTM to obtain the sentence context of each token. Next, each token representation is passed through two separate Feed-Forward Neural Networks (FFNN). The first is used to obtain a representation that the token is a start token, and the second is used to obtain a representation that the token is an end token. Finally, to classify these candidate spans into the possible entity types, they use a ranking to compute a score. To perform the multi-class classification step, they applied the following constraint to the computed scores: an entity is selected as long as it does not collide with the boundaries of higher-ranked entities.

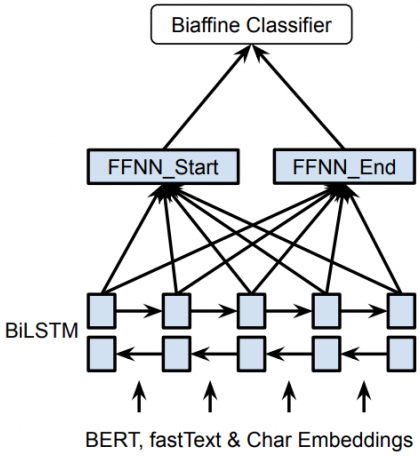


Figure 5.7: Overview of the Biaffine architecture.

We compared the MLC model with the Layered, Exhaustive, and Boundary approaches, as they performed well in both GENIA and GermEval. This is important as few papers have been tested on the German corpus due to the centralization of nested NER research on English resources. Considering baselines that include pre-trained language models, we reproduced the Recursive-CRF model as it belongs to the sequence labeling-based category like ours, thus facilitating an ablation study and hyperparameters comparison. Moreover, as shown in Table 5.2, it is one of the few architectures capable of addressing the three types of nesting. Finally, we replicated the Biaffine and Pyramid models since they are the current state-of-the-art models in GENIA.

Model	ME	NDT	NST
Layered	✓	✓	✓
Exhaustive	✗	✓	✓
Boundary	✗	✓	✓
Recursive-CRF	✓	✓	✓
Biaffine	✗	✓	✓
Pyramid	✓	✓	✓
MLC	✓	✓	✗

Table 5.2: Nesting types identified by the architectures used in our experiments. Multi-label entities (ME), nesting of different types (NDT), and nesting of the same type (NST).

5.3.2 Implementation Details

Pre-trained Word Embeddings

To encode sentences, we selected pre-trained word embeddings belonging to the same domain of each corpus. In experiments with GENIA, we used biomedical embeddings trained on MEDLINE abstracts [21]. For GermEval, we incorporated German FastText embeddings [38], and for the Chilean dataset, we used the medical pre-trained embeddings used in previous chapters. Again, we found that leaving the embeddings dynamic during training led to better results than leaving them static.

Contextual Word Embeddings

To study the impact of adding pre-trained language models in the embedding layer, in Table 5.3 we list the language models used for each corpus:

Corpus	BERT	Flair
GENIA	bert-large-uncased	pubmed-forward and pubmed-backward
GermEval	bert-base-german-uncased	de-forward and de-backward
Chilean Waiting List	bert-base-spanish-wwm-uncased	es-clinical-forward and es-clinical-backward

Table 5.3: Pre-trained language models used in our experiments.

Regarding the Biaffine model, the BERT embeddings were created using the paragraph-level context rather than sentence-level context. However, Fu et al. [37] explains that this method provides better performance in resolving correlations. Therefore, it is not an entirely fair comparison with baselines that use sentence-level context. For this reason, we do not make a comprehensive comparison with this model in terms of contextualized embeddings.

Parameters

We used a unified setting for all the experiments with MLC. The best hyperparameters were chosen by performing a random search over the range of values shown in Table 5.4, selecting

the best configuration based on performance on the development set.

To perform a fair comparison with baselines, we used the best hyperparameters reported in their papers. All the baselines were executed with the official code provided by the authors. To ensure reproducibility, the source code of our experiments is freely available in our repository[4].

Parameter	Range	MLC
max epochs	{20, 100, 150}	150, 100, 100
optimizer	{SGD, Adam, AdamW}	SGD
batch size	{8, 16, 32}	32, 16, 16
learning rate	{0.0001, 0.001, 0.1}	0.1
char emb dim	[20, 50]	25, 35, 25
dropout	[0.2, 0.8]	0.3, 0.3, 0.5
BiLSTM depth	{1, 2, 3}	3
BiLSTM hidden size	{128, 256, 512}	128

Table 5.4: Hyperparameter search space and the best values found for the MLC model. In the case of continuous intervals, 5 values were selected in the interval with the same distance. If three values are given, they represent the best values found for the GENIA, GermEval and Chilean Waiting List datasets, respectively.

5.3.3 Evaluation Metrics

We divided our metrics analysis for the models into the standard metrics already described in previous chapters and nested metrics proposed by us.

Overall Performance

Performance was evaluated using precision, recall, and micro F1-score over all entities in the test partition, the same metric used in previous chapters. One of the main drawbacks of using this metric is that a model that can recognize flat entities accurately but not nested entities will also have outstanding performance. In other words, since flat entities are much more common than nested entities, the above metric confounds flat and nested results and, consequently, cannot reflect well the ability of a model to detect nesting. To alleviate this issue, we analyze task-specific metrics proposed in previous work that adequately measure the model's ability to detect nested and non-nested entities.

Nested Performance

In our research about model's performance concerning the nesting cases, we were interested in studying four specific metrics: First, we wanted to know how well models can recognize

[4] `https://github.com/matirojasg/nested-ner-mlc`

entities that do not participate in nesting, better known as flat entities (m_{flat}). Then, we calculated the opposite case, i.e., measuring how well the models handle entities that participate in a nesting (m_{nested}). We consider an entity nested if it is nested within another entity or contains another entity. Finally, to analyze the score obtained with the m_{nested} metric, we calculated the ability of the models to detect inner entities (m_{inner}) and the outermost entities of a nesting (m_{outer}), which could provide us valuable information to improve the models in the future. Note that m_{nested} encompasses the m_{inner} and m_{outer} metrics.

However, none of these existing metrics capture the ability of the models to recognize both inner and outer entities simultaneously. For this reason, and to demonstrate whether the choice of a model in a dataset depends on the types of nesting present, we computed a score for nesting ($m_{nesting}$) and on the different types of nesting described in the task formalization (m_{ME}, m_{NDT}, m_{NST}). A nesting is considered correct if both inner and outer entities are recognized correctly.

In Figure 5.8, we can see an example of an annotation with the different cases we are measuring. First, we observe that there are no cases of flat entities or NST nestings. Secondly, we recognize two complete nestings: "INSUFICIENCIA CARDIACA CF II" and "HTA", where the first one corresponds to the NDT case, while the second one is a ME case. In both nestings, all participating entities are considered nested entities. In addition, "CF" is considered an inner entity, while "INSUFICIENCIA CARDIACA CF II" would be an outer entity. In the case of "HTA", since it belongs to the case of multilabel entities, both entities involved are considered inner and outer at the same time.

Figure 5.8: Example of an annotation in the Chilean Waiting List corpus to explain the different types of nesting.

The above metrics were calculated using precision, recall, and micro F1-score, but we only report the last one for brevity. We emphasize that most of these metrics have not been used before in nested NER research. Therefore, we believe it is crucial to incorporate them in future work as it allows us to measure and differentiate the performance of models on nested and non-nested entities.

5.4 Results

5.4.1 Main Results

Table 5.5 shows the overall performance of the proposed model against baselines on three different datasets. Despite its simplicity, we observe that the MLC architecture outperforms

Model	GENIA			GermEval			Chilean Waiting List		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
Layered	73.9	68.7	71.2	71.8	64.1	67.7	75.0	72.8	73.9
Exhaustive	74.1	69.7	71.8	78.6	64.6	70.9	76.3	71.7	68.2
Boundary	76.7	71.8	74.2	74.4	65.5	69.7	74.0	67.6	70.7
Wang et al. [124]†	-	-	-	74.8	70.5	72.6	-	-	-
Pyramid	78.1	72.8	75.3	77.8	66.9	71.9	79.6	75.4	77.5
Biaffine	79.1	73.7	76.3	89.0	77.4	82.8	81.5	67.1	73.6
Recursive-CRF	75.8	75.2	75.5	85.1	78.2	81.5	75.1	77.2	76.1
MLC	77.6	74.2	75.8	86.8	77.2	81.7	77.7	78.3	78.0
LM-based									
Dadas and Protasiewicz [24] [BERT + Flair]†	-	-	-	86.6	80.6	83.5	-	-	-
Luan et al. [77] [ELMO]†	-	-	76.2	-	-	-	-	-	-
Straková et al. [110] [BERT + Flair]†	-	-	78.3	-	-	-	-	-	-
Wang et al. [123] [BERT + Flair]	80.3	78.3	**79.3**	-	-	-	-	-	-
Biaffine [BERT]	79.9	76.5	78.1	88.3	85.0	86.6	78.7	70.8	74.5
Recursive-CRF									
- Flair	77.1	78.0	77.6	83.4	82.9	83.2	78.0	79.9	78.9
- BERT	76.4	77.4	76.9	84.3	83.0	83.6	76.6	77.8	77.2
- BERT+Flair	77.4	76.8	77.1	84.8	82.1	83.4	77.1	77.9	77.5
Pyramid									
- Flair	77.8	75.6	76.7	83.4	80.0	81.7	80.1	77.2	78.6
- BERT	79.1	76.9	78.0	87.7	85.8	**86.7**	78.0	73.6	75.7
- Flair + BERT	80.4	75.0	77.6	87.7	84.4	86.0	78.5	77.2	77.9
MLC									
- Flair	80.1	75.2	77.6	85.3	82.4	83.8	80.6	80.5	**80.5**
- BERT	79.4	74.3	76.8	85.1	80.3	82.6	79.7	78.8	79.3
- BERT+Flair	78.8	75.2	75.5	84.7	80.1	82.3	79.9	78.1	79.0

Table 5.5: Overall results on three nested NER corpora, including ours. † Indicates that scores are taken from the original papers. The rest of the experiments were reproduced by us. In addition, the "-" symbol means that there are no reported results for this corpus.

existing state-of-the-art models on the Chilean Waiting List by +1.6% in terms of the F1 measure. By contrast, although state-of-the-art is not obtained in GENIA and GermEval, we can see that MLC outperforms many specialized nested NER architectures, thus being a competitive approach. One possible reason for the excellent performance is that we use one model per entity type, which means that the number of possible labels is only one per model, avoiding the problem of nested entities and making the classification step more straightforward compared to other architectures. Compared with the statistics in Table 5.1, we can conclude that it is more challenging to obtain good results when the corpora have entities of a more considerable length. This can be explained by the strict metric we are using, where the boundaries and the entity types are requested to match.

We further analyze the effect of adding pre-trained language models in our experiments. As we believed, all models benefit from incorporating contextual word embeddings, improving their performance considerably compared to their base version. In GermEval, a general-purpose corpus, the language model that best improves the model's performance is BERT, while in the other corpora, it is Flair. Also, we can see that stacking Flair and BERT embeddings does not produce better results. We attribute this to the high dimensionality of these representations and to the fact that the two language models were trained on different corpora.

Regarding the Chilean corpus, which contains the highest percentage of nested entities,

we observe that the MLC model with Flair embeddings improves by +2.5% compared to its base version without pre-trained language models. This demonstrates the effectiveness of using Flair over BERT in this corpus. We suspect that it is due to the large number of misspelled and out-of-vocabulary words found in the unstructured clinical text. As pointed out in Akbik et al. [3], handling these types of words is one of the main advantages when using its character-level language model.

Despite the promising results, we hypothesize that benchmarking against the standard nested NER metric may not be a good indicator of model performance on nesting since most of the entities are not nested. Therefore, we analyze the results using nested metrics.

5.4.2 Nested Results

In most cases, the revisited nested metrics presented in Table 5.6 are relatively consistent with results in Table 5.5. This means that models which obtain state-of-the-art using the standard metrics also perform well according to these metrics. For example, in the Chilean Waiting List, the best model (MLC) achieves the best results according to the m_{flat}, m_{inner}, m_{outer}, m_{nested} metrics, which is a remarkable result considering the large number of nestings present in this corpus. Another observation is that, unlike the other datasets, in GENIA is more complex to recognize inner entities over the outermost ones. This finding could be helpful when designing future architectures for this corpus.

As expected, the models with better performance according to the standard metric are also associated with good results using the m_{flat} metric. This may not be a good indicator in the nested NER task since most of the entities in these corpora are not nested, and the proper performance on nestings is not reflected. This issue becomes much more evident when analyzing our proposed nesting metrics, presented in Table 5.7. We observe that the results are significantly lower than those for the previous metrics of Tables 5.5 and 5.6. This reveals the difficulty of correctly recognizing the nesting cases. One possible reason for this low performance is that these metrics are strict, as internal and external entities must be correctly predicted.

Although the selected baselines are designed to deal with nestings of the same type, their m_{NST} results in GENIA and GermEval are poor, while the results using the m_{NDT} metric are much higher. This suggests that NST is the most difficult case to identify for all models. Therefore, we believe that a model should not be prematurely discarded based on its limitation to handle a particular type of nesting. For example, although the MLC architecture cannot strictly identify the NST case in GENIA and GermEval, it obtains excellent results on the NDT case and the outermost entities involved in the NST. In contrast, concerning the m_{ME} metric, we note that the performance of the four models addressing this case is quite good, suggesting that it is not a complex case to recognize but still not taken into account when building nested NER models.

Another interesting point is that although our corpus has a smaller number of tokens compared to the other datasets, there is no correlation between the results obtained and the size of the corpus, as hypothesized in Section 1.1. As we can see in Tables 5.6 and 5.7, the factors that most affect the performance of these deep learning models are the different types

GENIA				
Model	m_{flat}	m_{nested}	m_{inner}	m_{outer}
Layered	73.2	62.3	42.9	79.8
Exhaustive	76.6	55.0	42.6	67.9
Boundary	77.4	59.5	42.0	75.6
Biaffine [BERT]	81.2	**65.8**	**49.3**	80.5
Pyramid [BERT]	81.1	65.2	46.1	**82.4**
Recursive-CRF [Flair]	**81.5**	62.3	46.9	77.4
MLC [Flair]	80.7	63.8	41.7	82.2

GermEval				
Model	m_{flat}	m_{nested}	m_{inner}	m_{outer}
Layered	68.8	60.9	62.0	59.7
Exhaustive	73.4	56.1	65.7	45.7
Boundary	70.9	54.5	54.1	55.0
Biaffine [BERT]	88.4	76.6	**78.1**	75.0
Pyramid [BERT]	**88.5**	**76.7**	77.3	**76.1**
Recursive-CRF [BERT]	85.5	73.0	74.9	71.0
MLC [Flair]	86.0	71.6	74.5	68.4

Chilean Waiting List				
Model	m_{flat}	m_{nested}	m_{inner}	m_{outer}
Layered	73.4	74.5	82.4	64.5
Exhaustive	71.7	63.8	71.5	53.4
Boundary	73.4	61.1	65.5	55.4
Biaffine [BERT]	76.2	72.5	75.2	69.2
Pyramid [Flair]	79.0	78.1	84.7	69.3
Recursive-CRF [Flair]	80.3	77.4	82.8	70.4
MLC [Flair]	**80.9**	**80.1**	**86.2**	**72.5**

Table 5.6: Results on nested and non-nested entities.

of nesting present in the corpus and the ability of the models to identify those cases.

Finally, we highlight that in the Chilean corpus where the state-of-the-art is reached, almost half of the complete nestings ($m_{nesting}$) are correctly recognized, which is a reliable indicator of our model performance on the nested NER task. These results suggest that the MLC architecture is the model that better suits our problem and also should be considered in future state-of-the-art comparisons due to its effectiveness. Besides, we argue that there is still much work to be done in nested NER, as most models fail to simultaneously recognize the internal and external entities of nesting, which is one of the main objectives of the task.

GENIA				
Model	$m_{nesting}$	m_{ME}	m_{NDT}	m_{NST}
Layered	26.2	-	41.7	9.7
Exhaustive	25.8	-	41.2	17.7
Boundary	26.6	-	40.5	17.8
Biaffine [BERT]	**34.5**	-	**51.9**	**22.9**
Pyramid [BERT]	33.4	-	49.5	20.9
Recursive-CRF [Flair]	31.5	-	49.1	19.4
MLC [Flair]	27.9	-	47.8	0

GermEval				
Model	$m_{nesting}$	m_{ME}	m_{NDT}	m_{NST}
Layered	37.3	-	40.4	16.2
Exhaustive	27.8	-	38.2	9.7
Boundary	21.2	-	25.5	7.8
Biaffine [BERT]	55.7	-	**64.3**	20.8
Pyramid [BERT]	**56.5**	-	63.8	21.4
Recursive-CRF [BERT]	51.1	-	58.9	**23.9**
MLC [Flair]	49.1	-	59.3	0

Chilean Waiting List				
Model	$m_{nesting}$	m_{ME}	m_{NDT}	m_{NST}
Layered	51.6	71.1	49.5	-
Exhaustive	28.4	0	41.7	-
Boundary	28.2	0	35.4	-
Biaffine [BERT]	41.1	0	55.1	-
Pyramid [Flair]	54.9	**73.7**	57.9	-
Recursive-CRF [Flair]	56.0	71.7	58.8	-
MLC [Flair]	**60.6**	72.5	**60.0**	-

Table 5.7: Our task-specific metrics. If columns have no results, it means that there was not a significant number of examples in the test partition.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis, we proposed a simple yet powerful architecture for recognizing nested entities in the Chilean Waiting List corpus. Specifically, we revisited the multiple LSTM-CRF (MLC) sequence labeling-based approach, which uses a single flat NER model per entity type. We compared its performance with several state-of-the-art architectures and three nested NER datasets. Our experimental results show that adding a character-level language model to the MLC architecture contributes to achieving state-of-the-art in our corpus.

In addition, to alleviate some gaps found in current evaluation metrics used for nested NER, we proposed new task-specific metrics that adequately measure the performance of models on nested entities. The results according to these metrics are low, especially when it comes to recognizing complete nestings. This finding shows that most nested NER models are better at identifying flat entities or part of nested entities, which is not the primary goal of the task. This demonstrates that there is still much work to be done on the nested NER task.

The results obtained suggest that the MLC architecture is the model that best suits the nested NER task in our corpus, demonstrating that the performance of this model is far superior to other state-of-the-art models. We hope that our study will help raise awareness in the research community that overlooking intuitive models and using only standard metrics when evaluating a new complex solution can be misleading and create an overly optimistic impression of the new solution's performance.

Regarding our case study, i.e., the Chilean Waiting List, we believe that the MLC model can be used for many studies to understand the high demand present in this system. For example, using the interface described in Chapter 4, we can support the recognition of new cases of psoriasis within the Waiting List [64], which could be extended to the detection of all diseases. In addition, telemedicine has been proposed as one of the solutions to reduce waiting times in the public health system [86], especially in times of pandemic. To correctly estimate the effect, it is necessary to summarize the suspected diagnoses and check which ones are suitable for telemedicine consultations. We believe that the use of our model can

speed up these tasks.

6.2 Future Work

Future directions include modifying the MLC architecture to improve performance for all nesting cases. For example, we could train separate models for outer and inner entities for each entity type to handle the case of nestings of the same type. We also plan to analyze two underexplored issues in the NER task: crossing-entities and discontinuous entities. The first corresponds to cases where entities are not fully nested in other entities, but there is an overlap, and the second case is when entities do not necessarily have consecutive tokens in the sentence.

In terms of developing NER models for this corpus, future work includes improving the recall score for procedures and findings due to the importance of identifying these entities. The low scores for Findings are mainly explained by the lack of agreement on the boundaries since it is an entity with a very large average number of tokens. Therefore, this is a complex task even for a specialist. In addition, it is interesting to note that the error analysis performed also helped us to identify inconsistencies in the annotations, which should be corrected in future work.

Finally, our annotated corpus has hierarchical entities (for example, test result and sign/symptom are part of the entity finding), and we plan to investigate the hierarchical nested NER using architectures as in Marinho et al. [81]. In addition, our corpus has attributes and relations which we have not addressed yet. Once we have a higher amount of annotated referrals, we plan to host a shared task to advance this corpus's multiple challenges.

6.3 Contributions

Besides this thesis, our work has contributed to two published articles listed below, plus a third article in the process of being published.

- Pablo Báez, Fabián Villena, Matías Rojas, Manuel Durán, and Jocelyn Dunstan. The Chilean waiting list corpus: a new resource for clinical named entity recognition in Spanish. In Proceedings of the 3rd Clinical Natural Language Processing Workshop, pages 291–300, Online, November 2020. Association for Computational Linguistics.
- Pablo Báez, Felipe Bravo-Marquez, Jocelyn Dunstan, Matías Rojas, and Fabián Villena. 2021. Automatic extraction of nested entities in clinical referrals in Spanish. Accepted in ACM Transactions on Computing for Healthcare.
- Simple yet Powerful: An Overlooked Architecture for Nested Named Entity Recognition (In the process of being published).

Bibliography

[1] Ankit Agrawal, Sarsij Tripathi, Manu Vardhan, Vikas Sihag, Gaurav Choudhary, and Nicola Dragoni. Bert-based transfer-learning approach for nested named-entity recognition using joint labeling. *Applied Sciences*, 12(3):976, 2022.

[2] Alfred V. Aho and Jeffrey D. Ullman. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ, 1972.

[3] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/C18-1139`.

[4] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL 2019, Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, 2019.

[5] Beatrice Alex, Barry Haddow, and Claire Grover. Recognising nested named entities in biomedical text. In *Biological, translational, and clinical language processing*, pages 65–72, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `https://aclanthology.org/W07-1009`.

[6] American Psychological Association. *Publications Manual*. American Psychological Association, Washington, DC, 1983.

[7] Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6: 1817–1853, December 2005. ISSN 1532-4435.

[8] Galen Andrew and Jianfeng Gao. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40, 2007.

[9] Osvaldo Artaza and Claudio A. Méndez. Crisis social y política en chile: la demanda por acceso y cobertura universal de salud. *Revista Panamericana de Salud Pública*, 44: 1, 03 2020. doi: 10.26633/RPSP.2020.16.

[10] Alexei Baevski, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli. Cloze-driven pretraining of self-attention networks. *arXiv preprint arXiv:1903.07785*, 2019.

[11] Pablo Báez, Fabián Villena, Matías Rojas, Manuel Durán, and Jocelyn Dunstan. The Chilean waiting list corpus: a new resource for clinical named entity recognition in Spanish. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 291–300, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.clinicalnlp-1.32. URL https://www.aclweb.org/anthology/2020.clinicalnlp-1.32.

[12] Darina Benikova, Chris Biemann, Max Kisselew, and Sebastian Padó. Germeval 2014 named entity recognition shared task. 2014.

[13] Darina Benikova, Chris Biemann, and Marc Reznicek. NoSta-D named entity annotation for German: Guidelines and dataset. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2524–2531, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/276_Paper.pdf.

[14] Daniel M. Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. Nymble: a high-performance learning name-finder. In *Fifth Conference on Applied Natural Language Processing*, pages 194–201, Washington, DC, USA, March 1997. Association for Computational Linguistics. doi: 10.3115/974557.974586. URL https://aclanthology.org/A97-1029.

[15] Leonardo Campillos-Llanos. First steps towards building a medical lexicon for spanish with linguistic and semantic information. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 152–164, 2019.

[16] Leonardo Campillos-Llanos, Ana Valverde-Mateos, Adrián Capllonch-Carrión, and Antonio Moreno-Sandoval. A clinical trials corpus annotated with umls entities to enhance the access to evidence-based medicine. *BMC Medical Informatics and Decision Making*, 21(1):1–19, 2021.

[17] José Cañete, Gabriel Chaperon, Rodrigo Fuentes, and Jorge Pérez. Spanish pre-trained bert model and evaluation data. In *to appear in PML4DC at ICLR 2020*, 2020.

[18] Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133, 1981. doi: 10.1145/322234.322243.

[19] Kai-Wei Chang, Rajhans Samdani, and Dan Roth. A constrained latent variable model for coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 601–612, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1057.

[20] Nancy Chinchor and Patricia Robinson. Muc-7 named entity task definition. In *Proceedings of the 7th Conference on Message Understanding*, volume 29, pages 1–21, 1997.

[21] Billy Chiu, Gamal Crichton, Anna Korhonen, and Sampo Pyysalo. How to train good word embeddings for biomedical NLP. In *Proceedings of the 15th Workshop*

on Biomedical Natural Language Processing, pages 166–174, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2922. URL https://aclanthology.org/W16-2922.

[22] Viviana Cotik, Darío Filippo, Roland Roller, Hans Uszkoreit, and Feiyu Xu. *Annotation of Entities and Relations in Spanish Radiology Reports.* 2017.

[23] Noa P Cruz Diaz, Roser Morante, Manuel J Mana López, Jacinto Mata Vázquez, and Carlos L Parra Calderón. Annotating negation in spanish clinical texts. In *Proceedings of the workshop computational semantics beyond events and roles*, pages 53–58, 2017.

[24] Sławomir Dadas and Jarosław Protasiewicz. A bidirectional iterative algorithm for nested named entity recognition. *IEEE Access*, 8:135091–135102, 2020. doi: 10.1109/ACCESS.2020.3011598.

[25] Hercules Dalianis. *Clinical Text Mining: Secondary Use of Electronic Patient Records.* Springer, 2018. ISBN 9783319785028. doi: 10.1007/978-3-319-78503-5.

[26] Mariona Delor, Antonia Martí, and Marta Recasens. Ancora: Multilevel annotated corpora for catalan and spanish. 01 2008.

[27] Li Deng and Dong Yu. Deep learning: Methods and applications. *Found. Trends Signal Process.*, 7(3–4):197–387, jun 2014. ISSN 1932-8346. doi: 10.1561/2000000039. URL https://doi.org/10.1561/2000000039.

[28] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

[29] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[30] Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998. doi: 10.1162/089976698300017197.

[31] Roberto Estay, Cristóbal Cuadrado, Francisca Crispi, Fernando González, Francisco Alvarado, and Natalia Cabrera. Desde el conflicto de listas de espera, hacia el fortalecimiento de los prestadores públicos de salud: Una propuesta para chile. *Cuadernos Médico Sociales*, 57(1), 2017.

[32] Hao Fei, Yafeng Ren, and Donghong Ji. Dispatched attention with multi-task learning for nested mention recognition. *Inf. Sci.*, 513:241–251, 2020.

[33] Jenny Rose Finkel and Christopher D. Manning. Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 141–150, Singapore, August 2009. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D09-1015.

[34] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 363–370, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219885. URL https://aclanthology.org/P05-1045.

[35] Joseph Fisher and Andreas Vlachos. Merge and label: A novel neural network architecture for nested NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5840–5850, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1585. URL https://www.aclweb.org/anthology/P19-1585.

[36] R. Florian, H. Hassan, A. Ittycheriah, H. Jing, N. Kambhatla, X. Luo, N. Nicolov, and S. Roukos. A statistical model for multilingual entity detection and tracking. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 1–8, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N04-1001.

[37] Yao Fu, Chuanqi Tan, Mosha Chen, Songfang Huang, and Fei Huang. Nested named entity recognition with partially-observed treecrfs, 2020.

[38] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL https://aclanthology.org/L18-1550.

[39] Baohua Gu. Recognizing nested named entities in GENIA corpus. In *Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology*, pages 112–113, New York, New York, June 2006. Association for Computational Linguistics. URL https://aclanthology.org/W06-3318.

[40] Stephen Guo, Ming-Wei Chang, and Emre Kiciman. To link or not to link? a study on end-to-end tweet entity linking. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1020–1030, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N13-1122.

[41] Dan Gusfield. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK, 1997.

[42] Jerry R. Hobbs, Douglas Appelt, John Bear, David Israel, Megumi Kameyalna, and Mabry Tyson. FASTUS: A system for extracting information from text. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*, 1993. URL https://aclanthology.org/H93-1026.

[43] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, nov 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

[44] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.

[45] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[46] K. Humphreys, Rob Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham, and Yorick Wilks. University of sheffield: Description of the lasie-ii system as used for muc-7. *Proceedings of the Seventh Message Understanding Conferences (MUC-7)*, 06 2001.

[47] Ander Intxaurrondo, Juan Carlos de la Torre, H Rodriguez Betanco, Montserrat Marimon, Jose Antonio Lopez-Martin, Aitor Gonzalez-Agirre, J Santamarıa, Marta Villegas, and Martin Krallinger. Resources, guidelines and annotations for the recognition, definition resolution and concept normalization of spanish clinical abbreviations: the barr2 corpus. In *SEPLN*, 2018.

[48] Yufan Jiang, Chi Hu, Tong Xiao, Chunliang Zhang, and Jingbo Zhu. Improved differentiable architecture search for language modeling and named entity recognition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3576–3581, 2019.

[49] Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. A neural layered model for nested named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1446–1459, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1131. URL `https://www.aclweb.org/anthology/N18-1131`.

[50] Arzoo Katiyar and Claire Cardie. Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 861–871, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1079. URL `https://www.aclweb.org/anthology/N18-1079`.

[51] J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. *GENIA corpus—a semantically annotated corpus for bio-textmining*, volume 19. 07 2003.

[52] Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics (Oxford, England)*, 19 Suppl 1:i180–2, 02 2003. doi: 10.1093/bioinformatics/btg1023.

[53] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[54] Igor Kononenko and Matjaž Kukar. Chapter 3 - machine learning basics. In Igor Kononenko and Matjaž Kukar, editors, *Machine Learning and Data Mining*, pages 59–105. Woodhead Publishing, 2007. ISBN 978-1-904275-21-3. doi: https://doi.org/10.

1533/9780857099440.59. URL https://www.sciencedirect.com/science/article/pii/B9781904275213500034.

[55] Jan A Kors, Simon Clematide, Saber A Akhondi, Erik M Van Mulligen, and Dietrich Rebholz-Schuhmann. A multilingual gold-standard corpus for biomedical concept recognition: the mantra gsc. *Journal of the American Medical Informatics Association*, 22(5):948–956, 2015.

[56] Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M Lowe, et al. The chemdner corpus of chemicals and drugs and its annotation principles. *Journal of cheminformatics*, 7(1):1–17, 2015.

[57] Vijay Krishnan and Christopher D. Manning. An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 1121–1128, Sydney, Australia, July 2006. Association for Computational Linguistics. doi: 10.3115/1220175.1220316. URL https://aclanthology.org/P06-1141.

[58] John Lafferty, Andrew Mccallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. pages 282–289, 01 2001.

[59] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781.

[60] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition, 2016.

[61] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1030. URL https://www.aclweb.org/anthology/N16-1030.

[62] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition, 2016. URL https://arxiv.org/abs/1603.01360.

[63] Lukas Lange, Heike Adel, and Jannik Strötgen. NLNDE: The neither-language-nor-domain-experts' way of Spanish medical document de-identification. *CEUR Workshop Proceedings*, 2421:671–678, 2019. ISSN 16130073.

[64] C Lecaros, J Dunstan, F Villena, DM Ashcroft, R Parisi, CEM Griffiths, S Härtel, JT Maul, and C De la Cruz. The incidence of psoriasis in chile: an analysis of the

national waiting list repository. *Clinical and Experimental Dermatology*, 2021. doi: doi:10.1111/ced.14713.

[65] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

[66] Yann LeCun, Y. Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015. doi: 10.1038/nature14539.

[67] Bing Li. Named entity recognition in the style of object detection, 2021.

[68] Jing li, Aixin Sun, Ray Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, PP:1–1, 03 2020. doi: 10.1109/TKDE.2020.2981314.

[69] Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. A unified mrc framework for named entity recognition, 2020.

[70] Salvador Lima, Naiara Perez, Montse Cuadros, and German Rigau. Nubes: A corpus of negation and uncertainty in spanish clinical texts. *arXiv preprint arXiv:2004.01092*, 2020.

[71] Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. Sequence-to-nuggets: Nested entity mention detection via anchor-region networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5182–5192, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1511. URL `https://www.aclweb.org/anthology/P19-1511`.

[72] Donald A. Lindberg, Betsy L. Humphreys, and Alexa T. McCray. The unified medical language system. *Methods of information in medicine*, 32(4):281, 1993.

[73] Todd Lingren, Louise Deleger, Katalin Molnar, Haijun Zhai, Jareen Meinzen-Derr, Megan Kaiser, Laura Stoutenborough, Qi Li, and Imre Solti. Evaluating the impact of pre-annotation on annotation speed and potential bias: Natural language processing gold standard development for clinical named entity recognition in clinical trial announcements. *Journal of the American Medical Informatics Association : JAMIA*, 21, 09 2013. doi: 10.1136/amiajnl-2013-001837.

[74] Liyuan Liu, Jingbo Shang, Frank F. Xu, Xiang Ren, Huan Gui, Jian Peng, and Jiawei Han. Empower sequence labeling with task-aware neural language model, 2017.

[75] Jason P Lott, Denise M Boudreau, Ray L Barnhill, Martin A Weinstock, Eleanor Knopp, Michael W Piepkorn, David E Elder, Steven R Knezevich, Andrew Baer, Anna NA Tosteson, et al. Population-based analysis of histologically confirmed melanocytic proliferations using natural language processing. *JAMA dermatology*, 154 (1):24–29, 2018.

[76] Wei Lu and Dan Roth. Joint mention extraction and classification with mention hypergraphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 857–867, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1102. URL https://www.aclweb.org/anthology/D15-1102.

[77] Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3036–3046, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1308. URL https://www.aclweb.org/anthology/N19-1308.

[78] Ying Luo and Hai Zhao. Bipartite flat-graph network for nested named entity recognition, 2020.

[79] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf, 2016.

[80] Montserrat Marimon, Jorge Vivaldi, and Núria Bel Rafecas. Annotation of negation in the iula spanish clinical record corpus. *Blanco E, Morante R, Saurí R, editors. SemBEaR 2017. Computational Semantics Beyond Events and Roles; 2017 Apr 4; Valencia, Spain. Stroudsburg (PA): ACL; 2017. p. 43-52.*, 2017.

[81] Zita Marinho, Alfonso Mendes, Sebastiao Miranda, and David Nogueira. Hierarchical nested named entity recognition. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 28–34, 2019.

[82] Diego Martinez, Haoxiang Zhang, Magdalena Bastias, Felipe Feijoo, Jeremiah Hinson, Rodrigo Martinez, Jocelyn Dunstan, Scott Levin, and Diana Prieto. Prolonged wait time is associated with increased mortality for chilean waiting list patients with non-prioritized conditions. *BMC Public Health*, 2019. doi: 10.1186/s12889-019-6526-6.

[83] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, page 188–191, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1119176.1119206. URL https://doi.org/10.3115/1119176.1119206.

[84] Maad Mijwil, Adam Esen, and Aysar Alsaadi. Overview of neural networks. 2019.

[85] Andrei Mikheev, Claire Grover, and Marc Moens. Description of the ltg system used for muc-7. In *MUC*, 1998.

[86] Ministerio de Salud de Chile. Estrategia Nacional de Salud para el cumplimiento de los Objetivos Sanitarios de la Década 2010-2020, 2011.

[87] Ministerio Secretaría General de la Presidencia. *Ley 20.285*. 2008. URL https://www.leychile.cl/Navegar?idNorma=276363&idParte=.

[88] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P09-1113`.

[89] Antonio Miranda-Escalada, Aitor Gonzalez-Agirre, Jordi Armengol-Estapé, and Martin Krallinger. Overview of automatic clinical coding: annotations, guidelines, and solutions for non-english clinical cases at codiesp track of clef ehealth 2020. In *Working Notes of Conference and Labs of the Evaluation (CLEF) Forum. CEUR Workshop Proceedings*, 2020.

[90] Antonio Moreno-Sandoval and Leonardo Campillos-Llanos. Design and annotation of multimedica–a multilingual text corpus of the biomedical domain. *Procedia-Social and Behavioral Sciences*, 95:33–39, 2013.

[91] Aldrian Obaja Muis and Wei Lu. Labeling gaps between words: Recognizing overlapping mentions with mention separators. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2608–2618, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1276. URL `https://www.aclweb.org/anthology/D17-1276`.

[92] Isar Nejadgholi, Kathleen C. Fraser, and Berry de Bruijn. Extensive error analysis and a learning-based evaluation of medical entity recognition systems to approximate user experience. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 177–186, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.bionlp-1.19. URL `https://www.aclweb.org/anthology/2020.bionlp-1.19`.

[93] A Névéol, H K Dalianis, G Savova, and P Zweigenbaum. *Clinical Natural Language Processing in Languages Other Than English: opportunities and challenges*, volume 9:12. 2018. doi: https://doi.org/10.1186/s13326-018-0179-8.

[94] Maite Oronoz, Arantza Casillas, Koldo Gojenola, and Alicia Perez. Automatic annotation of medical records in spanish with disease, drug and substance names. In *Iberoamerican Congress on Pattern Recognition*, pages 536–543. Springer, 2013.

[95] Maite Oronoz, Koldo Gojenola, Alicia Pérez, Arantza Díaz de Ilarraza, and Arantza Casillas. On the creation of a clinical gold standard corpus in spanish: Mining adverse drug reactions. *Journal of biomedical informatics*, 56:318–332, 2015.

[96] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages

8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[97] Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models, 2017.

[98] Mohammad Sadegh Rasooli and Joel R. Tetreault. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733, 2015. URL http://arxiv.org/abs/1503.06733. version 2.

[99] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[100] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[101] Arthur L. Samuel. Some studies in machine learning using the game of checkers. *IBM J. Res. Dev.*, 3:210–229, 1959.

[102] Erik F Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*, 2003.

[103] Fei Sha and O Pereira. Shallow parsing with conditional random fields. *Proceedings of HLT-NAACL*, 05 2003. doi: 10.3115/1073445.1073473.

[104] Dan Shen, Jie Zhang, Guodong Zhou, Jian Su, and Chew-Lim Tan. Effective adaptation of hidden Markov model-based named entity recognizer for biomedical domain. In *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, pages 49–56, Sapporo, Japan, July 2003. Association for Computational Linguistics. doi: 10.3115/1118958.1118965. URL https://aclanthology.org/W03-1307.

[105] Takashi Shibuya and Eduard Hovy. Nested named entity recognition via second-best sequence learning and decoding, 2019.

[106] Takashi Shibuya and Eduard Hovy. Nested named entity recognition via second-best sequence learning and decoding, 2020.

[107] Mohammad Golam Sohrab and Makoto Miwa. Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2849, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1309. URL https://www.aclweb.org/anthology/D18-1309.

[108] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

[109] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France, April 2012. Association for Computational Linguistics. URL `https://aclanthology.org/E12-2021`.

[110] Jana Straková, Milan Straka, and Jan Hajic. Neural architectures for nested NER through linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1527. URL `https://www.aclweb.org/anthology/P19-1527`.

[111] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014.

[112] György Szarvas, Richárd Farkas, and András Kocsor. A multilingual named entity recognition system using boosting and c4.5 decision tree learning algorithms. In *Discovery Science*, 2006.

[113] Zeqi Tan, Yongliang Shen, Shuai Zhang, Weiming Lu, and Yueting Zhuang. A sequence-to-set network for nested named entity recognition, 2021.

[114] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003. URL `https://www.aclweb.org/anthology/W03-0419`.

[115] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003. URL `https://www.aclweb.org/anthology/W03-0419`.

[116] Özlem Uzuner, Ira Goldstein, Yuan Luo, and Isaac Kohane. Identifying patient smoking status from medical discharge records. *Journal of the American Medical Informatics Association*, 15(1):14–24, 2008.

[117] Özlem Uzuner, Imre Solti, and Eithon Cadag. Extracting medication information from clinical text. *Journal of the American Medical Informatics Association*, 17(5):514–518, 2010.

[118] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[119] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967. doi: 10.1109/TIT.1967.1054010.

[120] Bailin Wang and Wei Lu. Neural segmental hypergraphs for overlapping mention recognition. In *Proceedings of the 2018 Conference on Empirical Methods in*

Natural Language Processing, pages 204–214, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1019. URL https://www.aclweb.org/anthology/D18-1019.

[121] Bailin Wang, Wei Lu, Yu Wang, and Hongxia Jin. A neural transition-based model for nested mention recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1011–1017, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1124. URL https://www.aclweb.org/anthology/D18-1124.

[122] Jing Wang, Mayank Kulkarni, and Daniel Preotiuc-Pietro. Multi-domain named entity recognition with genre-aware and agnostic inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8476–8488, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.750. URL https://aclanthology.org/2020.acl-main.750.

[123] Jue Wang, Lidan Shou, Ke Chen, and Gang Chen. Pyramid: A layered model for nested named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5918–5928, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.525. URL https://www.aclweb.org/anthology/2020.acl-main.525.

[124] Yu Wang, Yun Li, Hanghang Tong, and Ziye Zhu. HIT: Nested named entity recognition via head-tail pair and token interaction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6027–6036, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.486. URL https://aclanthology.org/2020.emnlp-main.486.

[125] Yongxiu Xu, Heyan Huang, Chong Feng, and Yue Hu. A supervised multi-head self-attention network for nested named entity recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14185–14193, May 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17669.

[126] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. Luke: Deep contextualized entity representations with entity-aware self-attention, 2020.

[127] Juntao Yu, Bernd Bohnet, and Massimo Poesio. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.577. URL https://www.aclweb.org/anthology/2020.acl-main.577.

[128] Juntao Yu, Bernd Bohnet, and Massimo Poesio. Named entity recognition as dependency parsing, 2020.

[129] Jie Zhang, Dan Shen, Guodong Zhou, Jian Su, and Chew-Lim Tan. Enhancing hmm-based biomedical named entity recognition by studying special phenomena. *J. of Biomedical Informatics*, 37(6):411–422, dec 2004. ISSN 1532-0464. doi: 10.1016/j.jbi.2004.08.005. URL https://doi.org/10.1016/j.jbi.2004.08.005.

[130] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification, 2016.

[131] Changmeng Zheng, Yi Cai, Jingyun Xu, Ho-fung Leung, and Guandong Xu. A boundary-aware neural model for nested named entity recognition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 357–366, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1034. URL `https://www.aclweb.org/anthology/D19-1034`.

[132] G. D. Zhou. Recognizing names in biomedical texts using mutual information independence model and svm plus sigmoid. *International journal of medical informatics*, 75 6: 456–67, 2006.

[133] GuoDong Zhou and Jian Su. Exploring deep knowledge resources in biomedical name recognition. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pages 99–102, Geneva, Switzerland, August 28th and 29th 2004. COLING. URL `https://aclanthology.org/W04-1219`.