UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

Trabajo de Fin de Máster en Ingeniería y Ciencia de Datos

Recognition of professions in medical documentation

Alfredo Madrid García

Dirigido por: Lourdes Araujo Serna

Raquel Martínez Unanue

Curso: 2022-2023: 1ª Convocatoria

A mi familia

A model is only as good as the data it is trained on - Unknown

All models are wrong, but some are useful - George Box

Resumen

El reconocimiento de entidades nombradas en historia clínica electrónica es un área del procesamiento del lenguaje natural que busca identificar y extraer información de datos médicos no estructurados para su posterior manejo. Actualmente, se estima que la mayor parte de la información relativa al paciente se encuentra almacenada de forma no estructurada. Bajo esta premisa, han surgido en los últimos años múltiples tareas colaborativas y modelos que facilitan la identificación de entidades de diversa índole como procedimientos médicos, enfermedades o información personal. Debido al desempeño de éstos, se ha planteado su uso en el contexto del brote pandémico producido por SARS-CoV-2, para la identificación de profesiones que puedan estar expuestas a un mayor riesgo de infección como el personal sanitario.

Por lo tanto, en el presente trabajo, se propone un sistema capaz de identificar conceptos relacionados con las profesiones, a destacar, la ocupación, la situación laboral y las actividades de los distintos actores que intervienen en el proceso asistencial como los pacientes, familiares, personal sanitario, y otros. El sistema planteado hace uso de un corpus público, MEDDOPROF, y un corpus especialmente anotado para este trabajo, MOD, así como de modelos pre-entrenados de aprendizaje profundo basados en transformadores. Concretamente, se usan modelos pre-entrenados con textos en español de ámbitos diversos; BETO, ALBETO y DistilBETO; y un modelo pre-entrenado con textos en español pertenecientes al dominio clínico basado en RoBERTa.

Tras la experimentación, se obtiene un valor de F1 de 0.664 en el reconocimiento de entidades relacionades con la ocupación, haciendo uso del modelo pre-entrenado con textos clínicos, y un valor de F1 de 0.742 en la identificación de los actores involucrados. Por último, el modelo con mejor rendimiento, el pre-entrenado con textos clínicos, se aplica para la detección de ocupaciones en historias clínicas electrónicas pertenecientes al Servicio de Reumatología del Hospital Clínico San Carlos (HCSC).

Con este trabajo se concluye: a) la idoneidad de los transformadores en el reconocimiento de entidades; b) la necesidad de conjuntos de datos correctamente anotados; c) la utilidad en la práctica clínica que tienen estos modelos para el reconocimiento de entidades relacionadas con ocupaciones.

Palabras clave: *Detección de profesiones, procesamiento del lenguaje natural, historia clínica electrónica,, inteligencia artificial, reconocimiento de entidades nombradas, aprendizaje automático, determinantes sociales de la salud, transformador*

Abstract

Named Entity Recognition (NER) in Electronic Health Record (EHR) is the area of Natural Language Processing (NLP) that seeks to identify and extract unstructured information in medical data for further management. Currently, it is estimated that most of the patient information is stored in an unstructured form. Under this premise, in recent years, multiple collaborative tasks and models have emerged to facilitate the identification of various types of entities such as medical procedures, diseases, or personal information. Due to their performance, the use of these models has been considered in the context of the SARS-CoV-2 pandemic outbreak, to identify professions that may be exposed to a higher risk of infection, such as healthcare workers.

Therefore, in the present work, a system capable of identifying concepts related to professions is proposed, to highlight the occupation, the work situation, and the activities of the different actors involved in the care process, such as patients, relatives, health staff, and others. Such a system uses a public corpus, MEDDOPROF, and a corpus specially annotated for this work, MOD, as well as pre-trained language models based on transformers. BETO, ALBETO and DistilBETO Spanish general-domain pre-trained models, as well as a Spanish clinical and biomedical specific-domain pre-trained model based on RoBERTa, are used.

After experimentation, an F1 value of 0.664 is obtained in the recognition of occupation-related concepts, using the Spanish clinical and biomedical specific-domain pre-trained model, and an F1 value of 0.742 in the identification of the actors involved in the care process. Finally, the best-performing model (i.e., the one pre-trained with clinical documents) is applied to electronic medical records belonging to the Hospital Clínico San Carlos (HCSC) Rheumatology Unit.

This work concludes: a) the suitability of transformers in named entity recognition problems; b) the need for correctly annotated datasets; c) the clinical usefulness of these models to recognise entities related to occupations.

Keywords: *Occupation detection, natural language processing, electronic health record, artificial intelligence, named entity recognition, machine learning, social determinants of health, transformers*

Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
BBPE	Byte-level byte pair encoding
BERT	Bidirectional Encoder Representations from Transformers
Bi-LSTM	Bidirectional Long-short term memory
BIO	Beginning-Inside–Outside
BioNER	Biomedical Named Entity Recognition
BPE	Byte-Pair Encoding
BRAT	brat rapid annotation tool
BSC	Barcelona Supercomputing Center
CDM	Common Data Model
CNN	Convolutional Neural Networks
CRF	Conditional Random Fields
CV	Cross-Validation
DL	Deep Learning
EHR	Electronic Health Record
ESCO	European Skills, Competencies, Qualifications and Occupations
GDPR	General Data Protection Regulation
GPT	Generative Pre-trained Transformer
HCSC	Hospital Clínico San Carlos
HIPAA	Health Insurance Portability and Accountability
IAA	Inter-Annotator Agreement
ICD	International Statistical Classification of Diseases and Related Health Problems
ILO	International Labour Organization
IO	In-Out
LLM	Large Language Model
LMs	Language Models
LSTM	Long-short Term Memory
mBERT	Multilingual BERT
MEDDOPROF	MEDical DOcuments PROFessions recognition shared task

MIMIC-III	Medical Information Mart for Intensive Care III
ML	Machine Learning
MLM	Masked-Language Modeling
MNAR	Missing Not at Random
MOD	More Occupation Data
n2c2	National NLP Clinical Challenges
NER	Named Entity Recognition
NIOSH	National Institute for Occupational Safety and Health
NLNDE	Neither Language Nor Domain Experts
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NN	Neural Networks
NSP	Next Sentence Prediction
ODH	Occupational Data for Health
OOV	Out-of-Vocabulary
PCA	Principal Component Analysis
PLM	Pre-trained Language Model
POS	Part of Speech
ProfNER	Identification of professions and occupations shared task
QoL	Quality of Life
RNN	Recurrent Neural Networks
SDOH	Social Determinant of Health
Seq2Seq	Sequence-to-Sequence model
SHAC	Social History Annotated Corpus
SODA	SOcial DeterminAnts
SVM	Support Vector Machines
TEMU	Text Mining Unit
TF-IDF	Term frequency – Inverse Document Frequency
WHO	World Health Organization
WWM	Whole Word Masking
XGBoost	Extreme Gradient Boosting
XLM-R	XLM-RoBERTa

Chapter 1

Introduction and Motivation

1.1 Research in context and motivation

According to the latest World Health Organization (WHO) reports, almost 2 million people die yearly from work-related diseases and injuries. Nineteen per cent of these deaths are due to occupational injuries. Different occupational risk factors, such as exposure to long working hours and workplace exposure to different agents, such as air pollution, asthmagens, carcinogens, ergonomic risk factors, and noise, are behind these figures [1]. In addition, the economic burden associated with work-related diseases and injuries is not negligible, affecting not only health systems but also employee productivity and well-being. This translates into direct and indirect health care costs, such as loss of productivity, lost wages, administrative expenses, sick leave and so on [2], [3]. Nowadays, prevention is one of the most effective weapons in fighting these diseases. For that reason, some agencies, such as European Agency for Safety and Health at Work (EU-OSHA), International Labour Organization (ILO) or National Institute for Occupational Safety and Health (NIOSH) support the prevention of work-related diseases and try to improve the lives of individual workers while minimising the costs of work-related illnesses and deaths [4].

The effect of occupation on health has been studied at multiple levels: mental health [5], [6], physical health [7], health inequality [8], self-rated health [9], Quality of Life (QoL) [10] and male fertility [11]. The occupation information in Electronic Health Record (EHR) can be helpful for occupational health surveillance, better health outcomes, prevention activities, identification of workers' compensation cases, and for providing intervention strategies [12], [13]. In fact, according to NIOSH, work history is considered a Social Determinant of Health (SDOH) (occupation is also considered a SDOH according to WHO [14]), which could ideally help healthcare providers. However, such information is poorly studied as a SDOH. In fact, some authors have pointed out that clinical decision-making and population health activities are rarely guided by work information [15]. In addition, the information relating to occupation is either not recorded routinely or is poorly captured within standard EHR systems [16], [17]. Proposals for characterising the whole occupational details have been made. For instance, NIOSH has suggested a classification into the following categories: occupation, industry, employment status, employer, work schedule, occupational injury, occupational exposures, and work-related.

As reported by some researchers, advances in incorporating occupational information in EHRs can lead to more informed clinical diagnosis and treatment plans, as well as more effective policies, interventions, and prevention strategies to improve the overall health of the working population [2]. These authors have highlighted multiple benefits of incorporating occupational information into the EHR: improve the quality, safety, and efficiency of care and reduce health disparities; involve patients and families in their health care; improve care coordination; improve population and public health; ensure adequate privacy and security protections for personal health information.

On the other hand, Natural Language Processing (NLP) has been proven useful in countless applications, including knowledge extraction and information retrieval, context disambiguation, data quality assessment, predictive models, and sentiment analyses [18], [19]. As the worldwide adoption of Electronic Health Record (EHR) has experienced steady growth in the last decade [20],

[21]; NLP techniques have also gained attention in the clinical setting due to their usefulness in discovering hidden information and patterns from unstructured free texts; and also due to their ability to transform unstructured text into structured data [22], among others [23]. In fact, it is estimated that more than 40% of the data in an EHR are stored as free text [24], and these unstructured embedded data have been shown to be useful in improving phenotyping performance [25]. However, it is not always easy or trivial to extract and process the text so that hidden information is unveiled and becomes available to use for further analysis. Even more important is to ensure that the information extracted is accurate and reliable [26]. Clinical narratives feature some particularities that can exacerbate the processing task and should be considered; some of them are discussed below and in [24]:

- Clinical records are prone to contain multiple and different hedges distinguishing negation, uncertainty, condition and conditional temporal, family history, and referred subject (patient or other), which harden and fuzzy the information retrieval and extraction tasks.

- Clinical records are usually written employing concise language, domain-specific terms, and also containing spelling errors, abbreviations and acronyms, a high number of alternate spellings, or multi-words. Furthermore, clinical notes can be highly unstructured, non-standardised, and of varying lengths, and text cohesion is not always guaranteed.

- Redundancy issues can appear in the clinical notes of chronic patients with a long follow-up period [27], as a result of copy-pasting actions.

The need to accurately capture occupation information is crucial for the provision of direct clinical care and for secondary uses such as patient risk stratification [28]. In fact, it has been shown how occupation information can be extracted into the OMOP Common Data Model (CDM) [29]. For example, medical specialities, such as rheumatology, that deal with work disability [30], can take advantage of characterising the patient's occupation. The detection of occupation mentions is also relevant for the de-identification of clinical documents, as these data are considered personally identifiable information [31], although it is not a sanctioned item by Health Insurance Portability and Accountability (HIPAA)[32]. Nevertheless, in the medical domain, the occupation detection task has not received as much attention as other Named Entity Recognition (NER) activities, such as the identification of qualifiers (e.g., speculation [33], negation [34], family history [35], temporal information [36]) or other tasks (e.g., de-identification, comorbidities recognition). Only in recent years two specific shared tasks for profession recognition have emerged. MEDical DOcuments PROFessions recognition shared task (MEDDOPROF), a Spanish-specific shared task for profession recognition in medical documents [37], and Identification of professions and occupations shared task (ProfNER), a Spanish-specific shared task for profession recognition and occupation in social media [38]. The importance of text mining of professions and occupational status goes beyond health care and epidemiological research, and it is also relevant in more diverse fields such as social services, competitive intelligence, human resources, legal NLP and even gender studies as stated in [39].

Encouraged by recent advances in NLP, Deep Learning (DL) architectures, and Pre-trained Language Model (PLM), we propose the use of transformers to detect occupations in clinical narratives. Briefly, these models are pre-trained on a vast amount of data in an unsupervised way to learn the general structure of a language, the vocabulary usage and the domain-specific terms. Then, the weights of the neurons comprising the model are updated using task-specific data.

1.2 Objectives

Hereafter, the main objectives of this Master's thesis are presented:

> **Objective 1**: To develop a system capable of detecting occupation mentions in clinical narratives

Objective 2: To develop a system capable of detecting to whom the occupation mentions of objective 1 belong

Objective 3: Evaluation of the systems developed in objectives 1 and 2 with a collection of real clinical notes from the Hospital Clínico San Carlos (HCSC) Rheumatology Service

MEDDOPROF, a shared task organised by the TEMU-BSC that focused on the recognition of occupations in Spanish medical documents and held in IberLEF/SEPLN 2021, is used as the evaluation framework that guides the development of this work.

1.3 Master's thesis structure

This Master's thesis dealt with the development of a system capable of identifying occupations, and whether they are related to specific persons, within clinical narratives in a shared task scenario. The chapters presented below address the different steps taken to achieve these goals.

Chapter 2 presents the main NLP related concepts that make it possible to understand the work carried out throughout this document, including transfer learning, transformers, and Bidirectional Encoder Representations from Transformers (BERT) among others.

Chapter 3 is intended to provide an overview of the different research articles in which occupation detection and other occupation-related tasks are the main objectives. The chapter begins by introducing the methodology used to retrieve the research studies. Then, a literature review is conducted. Subsequently, a brief description of the articles published in MEDDOPROF shared task is provided to highlight the different approaches used to solve the task by the different teams. Finally, a short review of other applications of transformers in Spanish clinical settings is provided.

Chapter 4 constitutes the materials and methods chapter. It starts by describing the MEDDOPROF corpus. The chapter continues with a description of the steps taken to build a corpus with new training data, More Occupation Data (MOD), and how these additional data were annotated. Finally, this chapter ends with an introduction to the tools and resources used to conduct the experiments presented in the next chapter.

Chapter 5 illustrates the proposed system architecture and development phases: pre-processing, training, and post-processing. Regarding the first phase, it is discussed how annotations from a corpus are handled to feed a transformer model. Regarding the training phase, the different hyperparameters considered in this work and how their values are set is explained. Finally, the post-processing steps required to perform the evaluation are introduced, namely token alignment, length of test sentences, and format conversion. In summary, this chapter is about how transformer-based models are built from scratch and trained with different hyperparameters to identify occupation mentions in clinical narratives.

Chapter 6 addresses the evaluation of the results/predictions obtained from the different models trained in the previous chapter. Firstly, an introduction to the evaluation metrics is made. Secondly, the results obtained from the two tasks are shown. Finally, an error analysis is performed to study the misclassification of the entities.

After all, *Chapter* 7 summarises the main ideas, findings, limitations and contributions of this Master's thesis. The chapter starts with a discussion of the different objectives of this work presented in this chapter, *Chapter* 1. An overview of the future trends and directions for extending this work is also conducted.

Chapter 2

Preliminary Concepts

This chapter defines the key concepts and theoretical aspects for understanding the work developed in this Master's Thesis. The description of the terms is aimed at understanding the architecture of vanilla transformers, and more concretely BERT. So, the definitions are particularised to these models. For further description and a better understanding of the concepts outlined here, the reader is referred to the following sources [40]–[53] and specially to Hugging Face webpage [54].

2.1 Natural language processing (NLP)

Natural Language Processing (NLP) is usually defined as the area of knowledge that emerges as the intersection of linguistics, computer science, and Artificial Intelligence (AI). The input of a NLP problem is natural language (i.e., as opposed to formal language, which has explicitly defined syntax and semantics), including both voice and textual data. Some of the most studied NLP applications are Part of Speech (POS) tagging, NER, text classification, sentiment analysis, text summarisation, machine translation, question answering, and speech recognition. In this work, NER is the leading actor.

2.2 Named entity recognition (NER)

Named Entity Recognition (NER) is commonly defined as *"an information extraction/retrieval sub-task that seeks to locate and classify named entities mentioned in unstructured text into pre-defined categories"* [55], [56]. Hence, the goal of a NER system is to identify all textual mentions of the named entities [57] (i.e., mentions of real-world entities, such as proper nouns. In practice, any mentions that are of interest to solve a user problem) from unstructured text and to classify them into pre-defined categories. The use of dictionaries (i.e., gazetteers), as shown in Section 4.2.3, that collect a list of all the possible entities is not always an option due to ambiguity and exhaustivity. In the biomedical domain, this term receives the Biomedical Named Entity Recognition (BioNER) name. The challenges of recognising biomedical entities have been stated by different authors in the literature [58]–[60]. Briefly, the non-standard usage of abbreviations; the presence of synonymous, homonyms, and ambiguities; highly specialised and technical terms and regular emergence of new ones; long entities such as chemical compounds and presence of control characters; terms combination; and sharing of nouns, hinder the entities identification task.

Traditionally, the methods applied to address such tasks are rule-based algorithms (e.g., pattern-matching techniques, heuristics), Machine Learning (ML) (e.g., Support Vector Machines (SVM), Conditional Random Fields (CRF)), and DL (e.g., Convolutional Neural Networks (CNN), Bidirectional Long-short term memory (Bi-LSTM), Recurrent Neural Networks (RNN)). The first one (i.e., rule-based algorithms), requires a considerable human effort to build a specific domain vocabulary able to capture most of the mentions, the results are highly dependent on the quality of the rules, have difficulties in dealing with negation, uncertainty, and ambiguity, and do not scale well with increasing data size and lacks flexibility and generalisability. Moreover, the risk of rules becoming outdated is always present because of the evolving nature of the biomedical language. The second one (i.e., ML), suffers from a similar drawback, handcraft features must be created in a time-consuming process and they require a high amount of labelled data to achieve good performance. The third one (i.e., DL), avoids the demanding steps of the previous approaches by automatically learning features from the data, although it is computationally expensive.

Nowadays state-of-the-art approaches focus on deeper architectures and pre-trained Large Language Model (LLM) that adopt self-attention mechanisms, such as transformers (see Section 2.3), which have shown superior performance compared to the rest of the approaches introduced above. Several research articles that implement the methods previously introduced have been conducted to identify medical-related entities in the biomedical literature [60], including the newer ones [61]. In this Master's thesis, the occupation detection is treated as a NER problem and therefore this concept plays an essential role in this work.

2.2.1 Tokenization

Tokenization is the process of dividing a string (i.e., sequence of characters) into individual tokens (i.e., sequence of tokens), commonly into words. In languages such as Spanish or English, white space characters could be used to identify word boundaries (conversely to other languages such as Chinese where the token is not separated by white spaces). However, there are cases in which other considerations must be taken into account and tokenization should be performed at the subword level to infer the meaning of new words from others with a similar linguistic construction (e.g., morphological derivation). Different tokenization techniques have recently been proposed, such as Byte-Pair Encoding (BPE), Byte-level byte pair encoding (BBPE), a variant of BPE called WordPiece tokenization, unigram tokenization, and SentencePiece tokenization. The vanilla transformer architecture implements the BPE tokenization, whereas BERT implements WordPiece tokenization as the mechanism to convert text into data that can be processed by the model. Briefly, this algorithm works by dividing each word in the training corpus into a sequence of letters. The initial vocabulary consists of the unique letters that form the words, distinguishing those letters from the starting letters of each word. All the existing pairs of letters in the corpus are listed by moving a shifting window to one position, and a score for each pair is calculated considering the frequency of appearance of the pair (i.e., the first and the last element). The pair with the highest score is selected, merged, and added to the vocabulary. This process is repeated until the desired vocabulary size is reached. Finally, once the vocabulary is defined, the tokenization process is conducted. A search for the longest possible token, in the vocabulary, contained in a word is conducted. Once located, the word is divided, and a new search begins until the word is completely split. To assess the performance of the WordPiece tokenizer, different performance metrics can be used, such as the average number of subwords produced per tokenized word or the proportion of tokenized words in a corpus that are split into at least two subtokens. More details on tokenization types can be found in the official Hugging Face page, and the WordPiece tokenization page. An example of how WordPiece tokenization works in our task is shown below:

"Trabaja en una instalación de atención a clientes en hostelería"

Is tokenize using BERT WordPiece tokenization to:

'tr', '##aba', '##ja', 'en', 'una', 'ins', '##tal', '##acion', 'de', 'ate', '##nc', '##ion', 'a', 'client', '##es', 'en', 'hostel', '##eria'

The double-hashtag (##) represents a prefix subtoken of the initial input. Each token shown above counts for the 512 subword token length limit of BERT model (more details on Section 2.4). The tokenization stage usually ends with the conversion of tokens to unique IDs (i.e., the learning process in Neural Networks (NN) is based on numbers), this is, to integer numbers. These IDs came from the corpus vocabulary used to pre-train the BERT model. Such vocabulary is fixed, so there is a chance that an unseen word coming from new data does not have its corresponding ID equivalence (i.e., Out-of-Vocabulary (OOV)). In this case, the special token [UNK] is assigned to those unseen words. The WordPiece tokenization helps to mitigate and reduce the appearance of [UNK] special tokens as it is more likely for a subword to appear elsewhere in the text than a whole word. This kind of tokenization reduces the number of words in the vocabulary (i.e., smaller embedding matrix), but fewer words will fit into a model that accepts a fixed number of tokens (such as BERT). So, there is a trade-off between the amount of information per token and the vocabulary size.

Ultimately, the tokenization process can be lossy with regard to the preservation of information. For instance, WordPiece tokenization separates punctuation characters. If an attempt is made to reconstruct the tokenised sentence, there is no certainty that the spaces between tokens will be preserved. In addition, tokens out of the vocabulary will receive the tag [UNK]. In a NER task or in a question-answering task, where the entity or answer span is relevant (i.e., the position of the entity in the text) to assess the performance of the model, special caution should be taken. As an example, if the following sentence is considered:

"El paciente tenía SARS-COV-2 y estuvo de baja"

And "baja" is an entity, the start-offset will be 42 and the end-offset 46.

After applying WordPiece tokenization and merging subtokens that start with ##, the remaining tokens will be:

['El', 'paciente', 'tenía', 'SARS', '-', 'COV', '-', '2', 'y', 'estuvo', 'de', 'baja']

If the previous tokens are used to reconstruct the original sentence (de-tokenized):

"El paciente tenía SARS - COV - 2 y estuvo de baja"

The "baja" entity start-offset and end-offset will be 46 and 50 respectively, so there exists an alignment shift.

2.2.2 Token representations: embeddings

The *embeddings* concept arises within the vector space (i.e., collection of vectors characterised by their dimension) and the vector semantic models. In these models, words are mapped to vectors, and those with similar meanings are close together in a multidimensional semantic vector space. Such space is usually defined by a four-element tuple (X, F, μ, β), to note:

- Vocabulary (X): set of tokens/strings that can be found in a text.
- Weighting function (F): projection of a text (i.e., sequence of tokens) into a multidimensional space.
- Similarity measure (μ): proximity between objects. Ideally, two texts with similar content should be found close together in the multidimensional space. Cosine similarity is commonly used as the similarity (i.e., semantic) measure. The angle between vectors gives an idea of the similarity, the higher the angle, the less similar the texts. This measure allows computing semantic similarity.
- Algebra (β): objects operators that facilitate mathematical operations (e.g., aggregation) over the representations.

Embeddings are representations of the meaning of words, this is, vectors that represent words are called embeddings. Often, this term is referred exclusively to short dense vectors (such as *Word2vec*, as opposed to sparse representations, such as Term frequency – Inverse Document Frequency (TF-IDF)), this is, real-valued numbers without a clear representation and with a vocabulary size much lower than the total number of words. The benefits of dense vectors include smaller parameter space promoting generalisation and avoiding over-fitting and fewer weights to learn. Two types of embeddings exist if considering the contextual information: Static (i.e., one fixed embedding for each word in a vocabulary) and contextualized/dynamic embeddings (i.e., the vector for each word is different depending on the surrounding tokens, this is, the context). Static embeddings are not appropriate when polysemy and homonymy phenomenon appears, as a word with multiple meanings is represented only in one way, irrespective of the context. In short, a static embedding is a function that maps each word type to a single vector (assuming a fixed vocabulary), typically this vector is dense with lower dimensionality than the size of the vocabulary. *Word2vec*, *GloVe* (i.e, capture global corpus statistics) and *FastText* (i.e., as Word2vec, able to handle unknown words) are examples of static embeddings. The underlying principles of static embeddings, such as Word2vec, involve training a binary classifier (i.e., multinomial logistic regression) to compute the probability that two words occur close together in the text by taking the learnt classifier weights and following a self-supervised approach.

Regarding contextual embeddings, each vector represents instances of a particular word in a particular context, this is, vectors representing some aspect of the meaning of a token in context. In the first case, static embeddings, a token has the same embedding irrespective of the context whereas in the second case, its embedding representation differs. Groundbreaking models such as Generative Pre-trained Transformer (GPT) or BERT take advantage of this kind of embeddings.

2.2.3 Segment representation: BIO (Begin, Inside, Outside) format

In this work, the occupation detection task is treated as a sequence-labelling NER task (i.e., to assign classes to an entire ordered sequence of tokens maximising the probability of assigning the correct classes to every token in the sequence, considering the sequence as a whole, not just as a set of isolated tokens [62]. Put in short: to produce some linguistic information per word). As a consequence, each token in a sentence is classified following a segment representation or chunk tag set. BIO tagging scheme [63] (also known as IOB2), where B stands for *first token in an entity*, I for *other tokens in an entity*, and O for *every token not included in an entity*, locates the boundaries of an entity in a sentence. BIO is proposed as the segment representation format due to its adoption by the research community and due to its use in BERT models as both input and output.

The BIO tags are followed by another tag that indicates the type of entity. Hence, this schema provides two kinds of tags: the position of an entity in a token and the category of the entity. Nowadays, different segment representation formats have been proposed (e.g., IO, IOB2/BIO, IOE2, IOBES, BI, IE, BIES). A comparison of them has been conducted elsewhere [64]. In this work, the entities' categories vary depending on the task. Table 2.1 shows the BIO schema particularised to the task of this project.

Table 2.1: Name entity recognition in MEDDOPROF tasks, represented with BIO schema

Sentence	El	paciente	es	deportista	profesional	en	activo
Task 1	O	O	O	B-PROFESION	I-PROFESION	O	O
Task 2	O	O	O	B-PACIENTE	I-PACIENTE	O	O

As discussed in the Tokenization section 2.2.1, the WordPiece tokenizer can split words into subwords. As an entity can be split into several subwords after tokenization, the labels of these subwords must be specified. Different alternatives for dealing with this issue exist: propagating the word's original label to all of its subwords, only labelling the first subword of each token, or creating an additional label for these cases. Table 2.2 shows this casuistry.

Table 2.2: Assigning labels to subwords

Sentence	de	##port	##ista	prof	##es	##sional
Task 1	B-PROFESION	?	?	I-PROFESION	?	?
Task 2	B-PACIENTE	?	?	I-PACIENTE	?	?

Lastly, it is important to note the difference between IOB1 and IOB2 formats, since some packages require the schema specification. In IOB1, B- is only used to separate two adjacent entities of the same type, whereas in IOB2, all entities begin with B-. See Table 2.3.

Table 2.3: IOB1 and IOB2 differences

Token	**IOB1**	**IOB2**
Es	O	O
abogado	B-PROFESION	B-PROFESION
de	I-PROFESION	B-PROFESION
familia	I-PROFESION	B-PROFESION

2.2.4 Active learning

The performance of any AI model is directly related to the amount of training data available and its quality. In NER tasks, the data have to be manually labelled by human annotators. Active learning pursues to reduce the label shortage problem by (i) strategically selecting which unlabelled samples to annotate, prioritising those that are supposed to have the greatest impact on the training. This is based on the premise that not all the labelled examples are equally important, and (ii) shifting the human annotation task to human correction task [65]. By selecting the notes that would have a higher impact on the training, the amount of labelled data required is decreased, and the training is speed-up. Those selected notes are the ones that the model is most confused about.

Regarding the selection of the samples to be labelled next, there are different strategies that rely on the *Query by uncertainty* concept. In this approach, an uncertainty score is assigned to all samples. Depending on this score, the algorithm chose to label or not a sample. Among the most common strategies inside this approach, the following stand out: least confidence (i.e., samples with the least confidence, $1 - Probability$, in their most likely label are selected), the margin of confidence sampling (i.e., samples with the smallest differences between the two most confident predictions are selected), ratio sampling (i.e., same as margin of confidence sampling but with ratios rather than differences), entropy sampling (i.e., samples with the highest Shannon's entropy are selected).

Regarding the simplification of the human task from annotation to correction, this is done through a four-step iterative process: manually annotating a small subset of data, training a model, pre-tagging/predicting the unlabeled samples with the model, and human verification and correction. This process is done iteratively for improving the model performance. Active learning approaches for clinical data have been presented in various studies [66].

2.3 Transformers

Transformers were born in 2017 with the publication of [67]. In this publication, the transformer model was originally intended for machine translation tasks, this is, mapping sequence of input vectors to sequence of output vectors (i.e., Sequence-to-Sequence model (Seq2Seq)). These models were rapidly accepted in the research community as they overcame some of the limitations of previous architectures (e.g., Recurrent Neural Networks (RNN) and Long-short Term Memory (LSTM)), such as the long-term dependency, becoming the state-of-the-art of several Natural Language Processing (NLP) applications, besides machine translation. Transformers are DL models, usually containing more parameters than other DL models such as CNN or RNN, that rest on three pillars: self-attention mechanisms, transfer-learning and an encoder-decoder module (the decoder is sometimes omitted, such as in BERT or the encoder, such as in GPT). This architecture avoids recurrent connections, is widely used in other DL approaches and combines linear layers, feed-forward networks, and self-attention layers. By removing recurrent connections, the vanishing and exploding gradients issues are avoided, the training requires fewer steps, parallelisation is easier, and longer-range patterns can be better captured [40]. In addition, effective scalability on parallel computing architectures can be achieved [52]. Transformers only rely on self-attention mechanisms for capturing the dependencies between the words in a sentence. In contrast to other architectures such as RNN, in which the input is sequential (i.e., a word input at a time), in transformers, the whole sentence is treated as input at one shot. A Spanish theoretical introduction to transformers is conducted in [68].

2.3.1 Attention mechanism and self-attention

An attention mechanism can be seen as a layer in a NN whose ultimate goal is to learn long-range global features, deciding which components of the input sequence contribute the most to the output [49]. This is, to assign a different amount of weight to each element in a sequence. The attention mechanism implemented in transformers is called self-attention.

Conversely to other structures, such as RNN, self-attention layers are able to extract information from large contexts without requiring intermediate layers. This kind of layer maps input vector sequences to output vector sequences of the same length. The model not only considers the current input/word (x_1), when computing its representation/embedding, but all the inputs above the actual one ($x_2, x_3, ..., x_n$), so it relates each word to all the other words in the sentence to have a better understanding of the actual word, see Figure 2.1. In addition, the computation of each sentence is independent of the rest, enabling parallelisation while training.

Self-attention relies on the representation of three vectors: *Query (Q), Key (K), and Value (V).* Starting from an input embedding matrix, X, in which each row is the embedding representation of each word in a sentence, and each column is an embedding dimension, three new matrices of the same size are created Q, K and V by multiplying X by three randomly initialised weight matrices W^Q, W^K, W^V (their weights are updated during training). Each row in the Q, K and V matrices contains the value vectors of each word. Once the Q, K and V matrices are computed, the dot product (which facilitates the preservation of the dimensions along the sublayers) between Q and K^T ($Q \bullet K^T$) is calculated. With this operation, a measure of how similar a word is to all the other words within a sentence is obtained. Then, the resulting matrix is divided by the square root of the dimension of the key vector for obtaining stable gradients, and normalised using the softmax function. This function compresses the values to [0-1] range and therefore the values can be interpreted as probabilities. Finally, the attention matrix Z is built by multiplying the previously calculated matrix by V. For a better understanding, the reader is referred to [42].

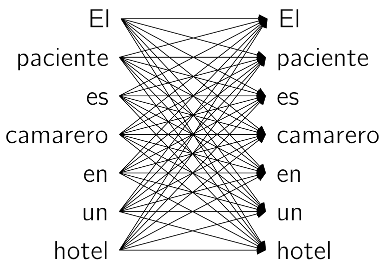

Figure 2.1: Self-attention mechanism fundamentals

The self-attention mechanism scales with quadratic complexity $O(n^2)$ with regard to the sequence length. Hence, long sequences make training time-consuming. This complexity limits the length of the context that Transformers can process. Efforts have been made recently to decrease this complexity and reduce training time [69].

2.3.2 Transfer learning and fine tuning

Training a transformer model from scratch is computationally expensive and sometimes unfeasible due to the amount of data needed (i.e., usually more than millions of sentences).

Transfer learning is a technique used in some DL domains such as computer vision that allows the reuse of most of the layers (generally low-level layers) of a NN trained on a specific problem, to improve generalisation in another setting. Hence, it is said that knowledge is transferred from one task, domain, and/or language (i.e., cross-lingual transfer) into another one providing a better initialisation (i.e., the weights of the new model are initialised with the weights of the old one instead of randomly). By using such a technique, the training in the new domain is speed-up, since the network does not have to learn from scratch, requires less training data and computing power, and costs are reduced. According to [70], the general framework for adapting pre-trained models in NLP involves the following steps:

1. Pre-training: in this step, transformer models like BERT are trained in an unsupervised manner using large-scale corpus and techniques such as language modelling (more details can be found in Section 2.4). The pre-training techniques vary from models.

2. Domain adaptation and fine-tuning: in this step, the model pre-trained in step 1 is adapted using the domain-specific corpus. Setting our work as an example, the pre-training can be conducted with Spanish general corpus (i.e., BETO) and the domain adaptation can be achieved using MEDDOPROF corpus. In short, the weights are adjusted to the new task.

Transfer learning in NLP can be seen as a way for the new model to understand the linguistic underlying mechanisms, thanks to external knowledge, to perform better in the task for which it was built. To sum up, transfer learning is used to re-use previously trained models, usually using general data corpora, that have learned general vocabulary, grammar, and word relationships. In Figure 2.2 a Transfer learning schema particularised to this Master's thesis objective is shown.

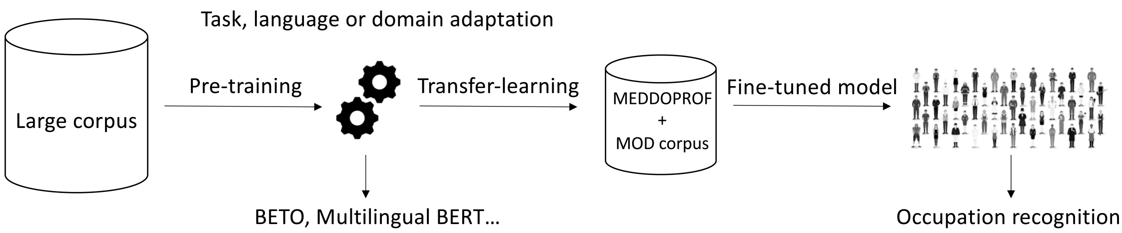

Figure 2.2: Transfer-learning in NLP

Due to the scarcity of annotated data, in many scenarios using pre-trained models via transfer learning is the only real choice. After that, a fine-tuning process, which is less data-intensive, is conducted using custom data. However, enough data of the classes to be predicted is still needed for optimal performance.

2.3.2.1 Sample size

There is no established consensus on the sample size required for NER fine-tuning on the target domain. Usually, a small number of annotated corpora difficult such a task. In [71], authors explored the effects of varying the training sample size in different humanity domains corpus. They showed that the performance of BERT models decreased when the number of target domain samples was reduced. Nevertheless, this drop was less pronounced when pre-training on the source domain and then fine-tuning on the target domain, compared to fine-tuning directly on the target domain. There is no rule to determine the minimum number of events required per entity. As a rule of thumb, some authors have pointed out a minimum of 200 training samples per label, and others, such as Microsoft, 50[1]. However, this size may be conditioned by different factors such as the number of labels to be recognized (i.e., as the number of labels increases, it may be harder for the model to be able to distinguish them, so the amount of data needed increases), the semantic proximity of labels (i.e., when the labels can be used in the same context interchangeably), the ambiguity between them or an unbalanced number of classes. In addition, the similarity of the domain problem to the original model pre-training data, and the complexity of the problem may increase or decrease the required number of training instances per class.

2.3.3 Vanilla transformer architecture

2.3.3.1 Encoder

Both the encoder and the decoder components can be any kind of NN architecture capable to model sequences. According to the original paper [67], the encoder is composed of a stack of N = 6

[1] https://learn.microsoft.com/en-us/azure/cognitive-services/language-service/custom-named-entity-recognition/faq

identical layers with each layer consisting of two sub-layers, a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. The higher the layer in the transformer stack, the more it learns and observes. Each layer passes on its knowledge to the subsequent layer. The final goal of the encoder is to transform the input into a representation that reflects the context, while paying more attention to the words that are more important to it. In other words, to convert a sequence of tokens into a sequence of embeddings (i.e., hidden state). It is important to note that the output of every sublayer has a constant dimension throughout the entire architecture (512 in the vanilla transformer and BERT). Below, the different components of Figure 2.3 are succinctly introduced.

1. Input embeddings: the input is fed into a layer for word embedding, converting the input tokens to vectors of dimension 512. In this step, the tokenization concept introduced in Section 2.2.1 is performed. BPE algorithm is the tokenization method implemented by the original transformer architecture. This layer is present only once in the encoder module.

2. Positional encoding: sine-cosine encoding based on the position of a word within a sentence. Each position is assigned a unique representation. This mechanism is analogous to the recurrence in RNN for tracking the position information of words. The output of the positional encoding layer is a matrix with each row representing an embedding of the sequence summed with its positional information. This addition gives each word a small shift in the vector space toward the position the word occurs in. Therefore, it is expected that semantically similar words that occur in near positions will be represented closer together in that space [43]. To sum up, the positional encoding function adds a value to the input embedding to describe the position of each token unequivocally.

3. Residual connection: transport the unprocessed input of a sublayer to a layer normalization function to preserve key information such as positional encoding.

4. Multi-head attention: integrates multiple self-attention modules allowing to associate each word in the input with the rest of the words in the same sentence for obtaining a better embedding for the word. More concretely, each attention sublayer contains eight heads followed by a post-layer normalisation head that adds residual connections to the output of the sublayer and normalises it. The results of the eight multiple attention heads are concatenated to obtain more robust results by calculating eight (in the vanilla transformer architecture) representation subspaces of how each word relates to the others and speeding up training. All the multi-head attention modules perform the same functions in all layers but look for different associations. Inside each head, the words are represented with the Q, K and V matrices presented in Section 2.3.3.1.

5. Post-layer normalization: layer that follows every attention and feedforward sublayers. This layer handles the residual connection that came from the input of the sublayer and is comprised of an addition function and a layer normalization (normalises each input in the batch to have zero mean and unity variance) that improves the performance of training. As the gradients can diverge with this approach, the learning rate is gradually increased during training (i.e., learning rate warm-up) [46].

6. Feed-forward network: this network contains two fully-connected layers and uses ReLU as the activation function. The most relevant aspect of this component is that it is a position-wise network, in which each position/embeddings is processed separately with the same operations, instead of processing the whole sequence of embeddings as a single vector.

2.3.3.2 Decoder

On the other hand, the decoder shares the same structure as the encoder (i.e., a stack of N = 6 layers, multi-head attention mechanism, and fully connected position-wise feedforward network layers), but adds a third layer, the masked multiheaded attention mechanism. A brief summary of its components is presented below:

1. Input embeddings: the input embeddings and the positional encoding are the same as in the encoder. Positional embeddings are transferred to the masked multi-head attention layer.

2. Masked multi-head attention: *masking* technique is applied to ensure that attention is only paid to the positions until the mask appears, forcing the transformer to learn how to predict, as future tokens are masked. This is, the decoder only "sees" words that come prior to the current word in the sentence. This is achieved using a look-ahead mask.

3. Post-layer normalization: same as in the encoder.

4. Multi-head attention: takes the output from the previous layers and combines it with the output from the encoder (i.e., dot product in attention operations).

5. Feed-forward network: same as in the encoder

6. Linear layer: produce the next probable element of a sequence, thanks to the softmax classifier that emits probabilities of an output.

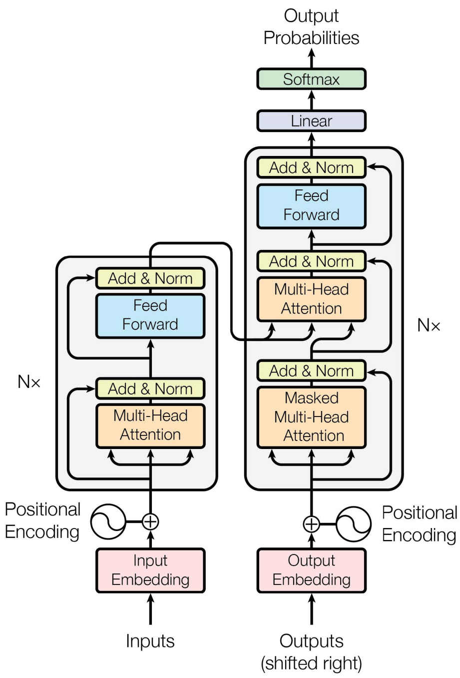

Figure 2.3: Transformers architecture. Source: *Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*[67]

Hence, transformers can be seen as NN that use dense vector feature representations that are context-sensitive, this is, the encoding is done considering the surrounding context of a given token. A comprehensive survey of this structure can be found in [72].

2.3.4 Transformers variants

Multiple variants of transformers, some of them trained with domain-specific data, have been created in recent years. Nowadays, more than 170 different transformers can easily be found in the Hugging Face webpage. Changes in the original transformer architecture (e.g., number of layers, heads, other positional representations such as relative positional representations, different tokenization methods, and so on) or in the pre-training tasks (e.g., dynamic masking, permutation language modelling) have led to the emergence of a wide variety of transformers capable of addressing all sorts of NLP problems. Depending on the encoder-decoder combination employed, three big families of models can be found:

1. Encoder-only: the input sequence is converted to rich numerical representations. Useful for NER and text classification. BERT is an example of such a model.

2. Decoder-only: models that predict the most probable word. GPT is the most well-known example of this family of models.

3. Encoder-decoder: both the input and the output are sequences. Useful for machine translation and summarisation tasks. BART and T5 are examples of this category.

Other categorisations can be made when considering the *language model* concept. Language Models (LMs) are statistical models that assign probabilities to sequences of words [47]. LMs have been defined as "*is a distribution $P(W)$ over the (infinite) set of strings in a language L*". This is, a probability distribution of a sequence of words. Language Models (LMs) can be classified into two categories depending on the context used to predict the next word:

1. Forward and backward autoregressive language modelling: unidirectional models that read words only in one direction to make predictions. They correspond to the decoder of the original transformer architecture. GPT is an example of autoregressive language models.

2. Auto-encoding language modelling: reads in both forward and backward directions. They correspond to the encoder of the original transformer. BERT is an example of autoencoding language models.

Most common transformer models fall into one of the next categories: autoregressive-models (e.g., GPT), autoencoding-models (e.g., BERT), seq-to-seq-models (e.g., BART), multimodal-models, retrieval-based-models (see Huggin Face for additional information). For simplicity, only the first two are considered in Table 2.4, which illustrates the context considered by them.

Table 2.4: Classification of transformers depending on the context taken into account: autorregresive and auto-encoding

Autorregresive	Forward	Trabaja en una instalación de atención a____
	Backward	____en hostelería
Autoencoding		Trabaja en una instalación de atención a____en hostelería

The most downloaded transformers in Hugging Face are usually (the statistics are collected each month): BERT base uncased/cased, BERT tiny, XLM-ROBERTa large and base, DistilBERT base uncased, Bio_ClinicalBERT, and ALBERT base.

For a transformers chronological timeline, the reader is referred to [73], for a recent overview on language models, the reader is referred to [74].

2.4 BERT

BERT is a bidirectional, pre-trained, autoencoding and context-based embedding model which was introduced in 2018 by Google Researchers [75]. It contains 12/24 encoder layers, 12/16 attention heads, 768/1024 hidden units (i.e., each token is represented as a 768/1024-dimensional vector) and 110/334 million parameters depending on whether it is *based* or *large* BERT (in scenarios where computational resources are limited, less complex models are preferred). It lacks the decoder module (and the corresponding masked multi-head attention sub-layers) of the vanilla transformer architecture and adds a bidirectional multi-head attention sub-layer. BERT typically uses the Adam optimiser with weight decay, the maximum number of tokens allowed is 512, and it was built considering the WordPiece tokenizer. The core idea of BERT is to use only the encoder, from the encoder-decoder module of the vanilla transformer architecture to transform the input into contextualized embeddings.

BERT introduces two major pre-training self-supervised tasks over the vanilla transformer architecture, one at the word level and the other at the sentence level.

1. Masking - Masked-Language Modeling (MLM): pre-training task that consists of masking a word (i.e., Whole Word Masking (WWM)) after tokenization (other approaches considered subword masking, rather than WWM), in a sentence with a selection probability of 15% (according to the original paper [75]). Then, the model is trained to predict the masked word. From each selected word, there is an 80% chance that the word will eventually end up being masked, a 10% chance of being replaced by a random word, and a 10% chance of remaining intact. This task can be seen as forcing the model to impute words in an incomplete sentence to better understand the particular use of language in a specific-domain context. Therefore, is a way to fine-tune specific-domain texts. Masked tokens are represented with the token ID 103 [MASK]. More details can be found in the Hugging Face webpage.

Trabaja	en	una	instalación	de	atención	a	**[MASK]**	en	hostelería
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
Trabaja	en	una	instalación	de	atención	a	clientes	en	hostelería

Whole word masking example

On the other hand, the original transformer architecture with the masked multi-head attention layer would have masked the rest of the sequence:

Trabaja en una instalación de atención a **<masked sequence>**

2. Next Sentence Prediction (NSP): pre-training task that aims to predict whether a sentence is the follow-up of a previous one or not, this is, to determine whether two sentences are consecutive. This pre-training task allows the model to "understand" the relation between sentences. In the 50% of cases, the following sentence is the actual following sentence of the previous one, in the other 50%, the following sentence is randomly selected.

   This pre-training task has been removed from recent transformers' architectures such as RoBERTa [76] as it is not as relevant as initially thought.

There exist other pre-training techniques implemented in other transformers such as causal language modelling or translation language modelling. Finally, a good introduction of BERT can be found in Hugging Face.

BERT relies on the multi-head attention mechanism, introduced in Section 2.3.3.1. With such a mechanism, the contextual representation (i.e., the embedding) of each word in a sentence is obtained by relating each individual word to all the words in the sentence, learning the relationship and contextual meaning of words.

2.4.1 BERT input data representation

In general, the input of a BERT model is converted into embeddings using the addition of three embedding layers.

1. Token embedding: embedding used to distinguish all the different tokens.
2. Segment embedding: embedding used to distinguish between consecutive sentences.
3. Positional embedding: embedding used to provide position information of each token to a model.

In Figure 2.4, a schema showing how the words of an input sentence are classified as entities can be appreciated. To begin with, since this is a NER problem, the segment embeddings and the position embeddings layers do not provide relevant information (conversely to what happens in sentence classification problems) and the input sentence is tokenised with the WordPiece tokenization algorithm. To continue, the [CLS] and [SEP] tokens also do not provide relevant information but are shown only for didactic purposes. The BERT's special tokens can be seen in Table A.3. The tokenised sentence goes through the encoder layers of BERT and the output is the embedding/representation of each token (i.e., the encoder is able to understand the context of an input sentence using a multi-head attention mechanism). Finally, the tokens are fed to a classifier comprised of a feedforward network and a softmax function. As the words were split into subtokens with the WordPiece Tokenizer, a decision has to be made regarding what is considered to be a recognised entity. For example, "Pac" could be considered the beginning of an entity (i.e., B-PACIENTE) while ##iente another special token, or the concatenation of "Pac" and ##iente could be the beginning of the entity. This is discussed in Section 2.2.3.

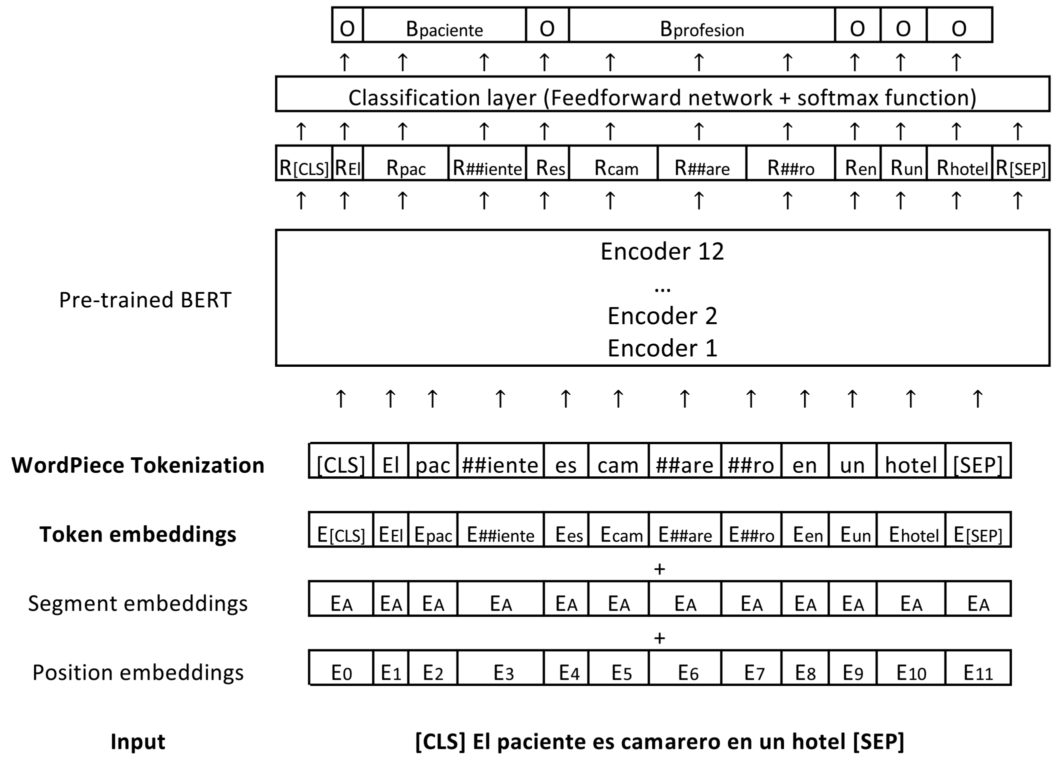

Figure 2.4: BERT applied to NER. B: Beginning of an entity, E: Embedding, 0: Outside, R: Representation of each token

2.4.2 Padding and truncation

Inserting non-informative elements into sentences of different lengths to homogenise and convert them into fixed-sized tensors, suitable as model input, is known as padding. A common approach is to add padding tokens into shorter sequences until they reach the longest sequence length and truncate them to the maximum sentence length accepted by the model (i.e., 512 subword tokens, approximately 300-400 words, for BERT, or 510 if adding the first [CLS] and last [SEP] special tokens, although it is not mandatory for NER tasks). Padding tokens are commonly represented by the [PAD] token.

A significant challenge arises when dealing with texts that exceed the maximum length limit of BERT. By default, the original BERT implementation automatically truncates longer sequences, with the consequent loss of information. As a general rule, the longer the sequence entered, the more context the model has, so using whole clinical notes could be useful for disambiguation. Truncation is usually done by removing end tokens (i.e., keeping the given number of subwords from the left). However, recent research seems to suggest that cutting in the middle of sentences longer than 512 subword tokens, rather than at the beginning or end, could have better performance in tasks such as text classification [77].

Other approaches capable of handling text inputs longer than 512 subword tokens have emerged recently, such as Longformer or Reformer [78], however, training time can be increased. In addition, researchers have proposed splitting sentences larger than the maximum length of the model, at the expense of losing some of the context; or using a sliding window, at the expense of higher computational costs. In both scenarios, context information can be lost due to sentence cutting at an arbitrary position. More details on padding and truncation can be found in the official Hugging Face documentation. Details about how to handle long texts can be found in [79]–[81]. Finally, a review of pre-trained language models for long clinical text is conducted in [82].

2.4.3 Attention mask

The attention mask is a binary mask that prevents the model from performing attention to padded tokens, this is tokens, without information, by setting a zero value to their positions, see Table 2.5. This is useful to ensure that the padding values (highlighted in red) are not processed along with the actual input values.

Table 2.5: Attention mask. In blue, attended tokens. In red, not attended tokens. Sentences 1, 2, and 3 are padded to the maximum length (i.e., length of sentence 4). Subword tokens are converted to unique IDs (i.e., numbers), which will feed the model.

Sentence	**Input Ids**									**Attention mask**								
1	101	49	2	74	82	0	0	0	0	1	1	1	1	1	0	0	0	0
2	101	13	42	36	125	32	13	0	0	1	1	1	1	1	1	1	0	0
3	101	23	5	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
4	101	83	91	51	37	287	384	82	102	1	1	1	1	1	1	1	1	1

101 and 102 token IDs corresponds to [CLS] and [SEP] token respectively

2.4.4 BERT variants

Multiple BERT models have been developed over time. These variants differ in the corpus selection they were trained in (i.e., domain-specific or general), the number of parameters, or present tweaks to the basic model. Three large families are presented below. Transformers' models for these families could be helpful and therefore considered for our work (and as shown in Section 3.4.1, some of them were considered and implemented by the different participant teams):

- Multilingual and monolingual BERT models: BERT models that acquired generalisability across languages as they are trained with a large corpus of multilingual data, or models that are trained on specific languages data. BERT Multilingual is one of the most widely used. There are approaches like zero-shot where a model is trained only with documents in a language and then fine-tuned/evaluated with documents in other languages. On its behalf, BETO is a BERT based model trained on a large Spanish corpus of similar size to BERT, consisting of a vocabulary of 31k BPE subwords constructed using the SentencePiece tokenizer. Trained with a general domain corpus [83], using the WWM technique, BETO has become one of the preferred models for Spanish NER tasks and the model taken as baseline.

- Domain-specific BERT models - clinical models: BioBERT [84], ClinicalBERT [85], MedBERT, DischargeSummaryBERT, PubMedBERT, or BioClinicalBERT [86] are examples of pre-trained BERT models in large-scale English biomedical corpora. The advantages of training a BERT model from scratch on a domain-specific corpus are: (i) the model learns specific embeddings of a domain, and (ii) the model learns the domain-specific vocabulary. In healthcare applications, they have shown better performance than the vanilla BERT implementation.

- Distilled BERT models: BERT models that follow the distillation paradigm, defined as: *"a compression technique in which a compact model - the student - is trained to reproduce the behaviour of a larger model - the teacher - or an ensemble of models."* [87]. Lighter, faster, cheaper, and smaller BERT models are created as the result of distillation, such as DistilBERT, by distilling BERT base while reducing the number of parameters.

Some BERT models, such as BETO can be found in two variants: *cased* and *uncased*. The cased variant does not lowercase capital letters in a word nor remove accents, so the input text remains unchanged. However, in the uncased BERT model, the text is lowercase before tokenization and the accents are not preserved. Cased BERT models are recommended when there is a high chance for an entity of study to be capitalised (e.g., Names, countries, brands and so on). Uncased BERT is generally preferred if the application is not sensitive to case information.

In addition, two BERT models that differ in the configuration size were initially proposed, BERT base and BERT large. Both differ in the number of blocks, hidden size, heads, and parameters.

A comparison of four BERT models can be found in Table A.7.

2.5 Spanish transformers models and applicability to NER

The use of transformers models pre-trained on Spanish text, and more specifically on clinical notes is desirable since the model learn both generalities and particularities of the Spanish clinical domain improving the performance. In this section, models that could be applied in a NER task are shown.

The most extended general-purpose BERT Spanish models are dccuchilebert-base-spanish-wwm-uncased and dccuchilebert-base-spanish-wwm-cased.

However, the first Spanish biomedical and clinical transformer-based pre-trained language model was presented in [88], and is RoBERTa-based. The model is accessible through Hugging Face, bsc-bio-ehr-es. Fine-tuned versions of this model have also been published for specific tasks, bsc-bio-ehr-es-pharmaconer, roberta-es-clinical-trials-ner, bsc-bio-ehr-es-cantemist, Spanish_disease_finder. Other BETO fine-tuned models to the clinical setting is beto-prescripciones-medicas. Finally, a fine-tuned version of the multilingual XLM-R transformer model, xlm-roberta-large-spanish-clinical, was also presented [89].

Other models trained with Spanish data not limited to the medical domain have been recently presented. RigoBERTa [90], based on DeBERTa, which outperformed the previous state-of-the art, MARIA [91] or BERTIN [92].

Chapter 3

Review of the State-of-the-Art

This chapter presents an overview of the different approaches used to characterise and identify occupation-related entities in free-text narratives (e.g., clinical notes, discharge letters, and emergency reports). Firstly, the search strategy is presented, secondly, a literature review is conducted to study the different tasks in which occupation was the main agent. Finally, the proposals submitted to the MEDDOPROF shared tasks and published are reviewed.

3.1 Search strategy and data collection methodology

A literature search was conducted to identify publications related to occupation characterisation and detection tasks in free-text narratives. The search was conducted in PubMed and Google Scholar. In addition, a review of the grey literature was performed. The keywords used were a combination of the following ones: *occupation, work-disease, profession, electronic-health record, named-entity recognition, natural language processing, identification, and detection, social determinants of health.* Some examples of the queries executed in PubMed can be shown below:

(("occupation information") OR ("work history")) AND ("electronic health record") AND ((identification) OR (detection))

(("occupation information") AND ("electronic health record")

(("social determinants of health") AND ("corpus") AND ("occupation" OR "employment")

MeSH terms were initially considered but not included. No filters were used for the search (e.g., article type, publication date, or language).

3.2 Occupation detection in the medical field

Occupation detection in the medical field has gained special attention in the last two years, since 2021.

The authors of [16] presented a 10-step method for developing and validating an application to text mining occupations from the free text of psychiatric clinical notes. To begin with, an interdisciplinary team developed annotation guidelines and annotated 600 personal history documents from a repository of de-identified clinical data. The annotation process was split into two parts: the occupation annotation itself and the occupation relation (i.e., the person with the occupation). This annotation was made as a training exercise, and then, 1,000 personal history documents were annotated, serving as a gold standard. From this point on, two different approaches were followed to identify occupations in the clinical narratives: a rule-based approach and a hybrid approach combining ML (e.g., Conditional Random Fields (CRF)) and rules. The authors distinguished the following implementation steps: A) text pre-processing: English tokeniser, lemmatise, sentence splitter, Part of Speech (POS) tagging, named entity transducer; B) occupation mention detection; C) occupation title assignment; D) occupation relation extraction; E) occupation filtering. In view of the results obtained, the authors achieved better precision performance when using the hybrid approach. An interesting discovery of this study was that the percentage of patients with an occupation recorded increased from 14% to 57% when considering unstructured fields.

Meanwhile, researchers from Manchester developed a large occupation dictionary used to identify occupation mentions [31]. The system design was evaluated on public and non-public clinical datasets from different institutions and countries. The workflow proposed by the authors consists of a pre-processing step with GATE and OpenNLP for tokenization, sentence splitting, POS and shallow parsing or chunking; followed by two components: knowledge-driven (dictionary and rules) and data-driven (ML). The first component was made up of a dictionary that contained 19,148 lexical entries with case insensitive and longest match to tag occupation mentions; and a rule-tagger which included a set of rules to restrict or reinforce the dictionary tagger to relevant sub-sections and specific contexts. The aim of the second component, the data-driven method, was to extract features from the preceding components to tag token sequences using a CRF tagger. The feature extraction consisted of different lexical, orthographic, contextual, and semantic features. Regarding the CRF tagger, the BIO token-level representation schema was used. The authors conclude that incorporating a large dictionary as part of a data-driven pipeline could help beat the previous state-of-the-art performance.

Social media occupation and profession recognition tasks have also been recently addressed, in 2021, in a shared task called Identification of professions and occupations shared task (ProfNER). This shared task focused on Spanish tweets data.

Although it has not been addressed as an individual task, occupation detection has been addressed simultaneously with other SDOH. A review published in 2021 [93] covered the approaches used for extracting SDOH. Of 6,402 publications, 82 met the inclusion criteria. Only seven articles included occupational information, and all of them consisted of rule-based algorithms. The occupation extraction task was commonly addressed together with education and smoking status extraction.

Authors from [94] built an annotated corpus, Social History Annotated Corpus (SHAC), using notes from MIMIC-III [95] and a dataset from the University of Washington. This corpus encompassed 4480 social history sections, and up to 12 SDOH entities were considered (i.e., substance use, physical activity, insurance, living status, and so on). The employment situation was studied considering the following tags: status (employed, unemployed, retired, on disability, student, homemaker), duration, history, and type. An event extraction model, based on an active learning framework with Bi-LSTM and CRF layers, was built, achieving a 0.81–0.86 F1-score for employment status. This corpus was used in the National NLP Clinical Challenges (n2c2) 2022 task 2 [96], whose results will be available during 2023[1]. For instance, the authors in [97], achieved a 0.88 F1-score when extracting the SDOH using a BioClinical-BERT-based model.

Authors from the United States, [98], built a corpus consisting of 4,063 clinical note sentences (originating from the clinical data warehouse at the University of North Carolina Health System) and six labels related to financial resources and poor social support. Employment and income insecurity were among them. Five classification models were trained, including Bi-LSTM models. The best performing, Extreme Gradient Boosting (XGBoost), achieved a 0.80 F1-score in the employment identification task.

Researchers from [99], developed annotation guidelines, annotated 2,670 MIMIC-III notes (as [94]) with 13 SDOH categories, and trained CNN, LSTM and BERT models. The occupational entities account for the 5.5% of the total, and the best model, BERT, achieved a 0.77 F1-score.

A recent article [65] explored the use of Bi-LSTM, CRF and BioBERT [84] to extract 10 SDOH (e.g., disease, gender, employment, relationship status and so on) from case reports of COVID-19 patients. Two hundred case reports were initially annotated by experts, and then, the authors followed an active learning approach. Finally, around 280k entities were annotated, and a 0.78

[1] https://academic.oup.com/jamia/pages/cfp-social-determinants

F1-score was obtained for employment detection.

Eventually, a NLP package called SODA, that included pre-trained transformers for extracting 19 SDOH categories (i.e., employment, language, financial constraint, sexual activity) for cancer patients (i.e., breast, lung, colorectal) was released by [100]. For that purpose, the researchers built a corpus of 629 patients and 13,193 entities. The number of annotated occupation concepts was 499, and four transformer models, including BERT and RoBERTa were trained.

3.3 Other occupation-related tasks

The representation of occupation information was addressed in [28]. In this study, researchers used six clinical sources to analyse free text mentions of occupation and related information within notes. With this in mind, they developed annotation guidelines derived from the NIOSH ODH model [101] (i.e., model that illustrates relationships and attributes for a person's employment status, retirement dates, past and present jobs, usual work [15]) and used brat rapid annotation tool (BRAT) to annotate the corpus. Five parent categories were considered: *occupational history, usual occupation, employment status, occupational injury*, and *occupational exposure*. Finally, 2,005 annotations from 868 sentences were mapped to 41 entities, and the frequency of the entities was characterised. The study's main purpose was to inform occupation representations, therefore, it lacked a system for recognising entities after performing the annotation and building the annotated corpus. The authors concluded that standardising the entry of EHR occupation information would improve data quality.

The quality of SDOH, including race, language preference, health insurance status, country of origin, socioeconomic status, level of education, environmental health and occupation, in EHR has been reviewed in [102]. Of 76 articles, seven studied the quality of occupation data, six examined data completeness, four found that the data was not Missing Not at Random (MNAR) and that female patients tend to have fewer occupation data in their EHR, and finally, one of them tried to impute occupational data. In this review, authors discussed [103] study, as an example. In this last research, the authors studied the availability and accuracy of occupation data in oncology firefighters' patients. Of almost 4,000 patients, only 17% have a firefighting-related code.

In another study, researchers studied the content and quality of free text occupation documentation in the EHR [17]. The authors proposed a five-level categorisation of data quality issues for occupation entries: *misspelling, acronym/abbreviation, ambiguous information, multiple entries/occupations, and other grammar-related issues.* The results of this study highlighted significant issues regarding the quality of occupation data and their low utility for secondary purposes, such as research, policy, or population initiatives.

In addition, the researchers of [104], generated a corpus from six distinct clinical sources, identified 868 occupation-related sentences, and annotated 2,005 entities. Some of the annotated entities were: occupational history, usual occupation, occupation status, occupational injury, occupational exposure, and occupational conditions. The objective of this study was to demonstrate to what extent occupation-related information within EHR can vary. On the other hand, a narrative review addressing how NLP has been successfully applied in occupational exposome research was published [105]. In this context, the exposome was defined as "*the measure of all the exposures of an individual in a lifetime and how those exposures relate to health. An individual's exposure begins before birth and includes insults from environmental and occupational sources*" [106]. From an initial number of 6,420 articles, the authors reviewed 37 articles in-depth, making a distinction between ML and knowledge-based methods.

Eventually, occupation detection has been addressed in fields other than medical [107], [108].

3.4 MEDDOPROF shared task

MEDDOPROF shared task arose in 2021 as the "*The first shared task focusing on automatic recognition of professions and occupational status (and normalisation to standard multilingual terminologies) in medical documents*" [39]. The creation of this shared task was motivated by the COVID-19 pandemic outbreak as certain occupational groups (e.g., physicians, nurses, hospital cleaners, shopkeepers, geriatric caregivers, essential workers and those with higher degrees of social interaction) had an increased risk of mortality and morbidity [109], [110]. Furthermore, the shared task organisers also highlighted the relevance of characterising patients' professions for targeted vaccination plans. This task was defined as follows:

> *'The MEDDOPROF Shared Task tackles the detection of occupations and employment status, as well as their normalisation or entity mapping, in clinical cases in Spanish from over twenty specialities (i.e., psychiatry, internal medicine, oncology and so on)* [37].

MEDDOPROF was divided into three sub-tasks:

- Named Entity Recognition (NER): according to [37], this task pursues to *find exact mentions of occupations in the text and label them according to the type of occupation: profession (i.e., paid occupations), activity (i.e., non-paid occupations) or working/employment status (i.e., occupational + socioeconomic status)*. This is, the identification (beginning and end of an entity) and classification (profession, working/employment status, activity) of occupation mentions.
- **CLASS**: identification of the person to whom the occupation belongs (*patient*, *family member*, *health professional*, or *other*). This task can be seen as an extension of the previous one.
- **NORM**: according to the organisers, this task pursues to *enable semantic interoperability, data integration and practical exploitation of NER text mining systems*. This is expected to be achieved by normalising the detected entity mentions to European Skills, Competencies, Qualifications and Occupations (ESCO) and some SNOMED-CT codes. In short, this task is about occupation normalisation according to a reference code list.

The results of this task were presented in the IberLef2021 workshop, part of the SEPLN 2021 Conference. An overview paper of the MEDDOPROF shared-task was also published [37].

3.4.1 MEDDOPROF submitted works

The submitted works can be seen in Youtube. Fifteen teams from six countries and eight papers emerged as a result of the three sub-tasks. Different methodologies were applied by the participant groups: CNN (1), transformers (5), CRF (4), non-neural (2), RNN (1), attention mechanism (1). The software used varied between teams: CRFsuite, Keras, spaCy, scikit-learn, PyTorch, Huggyn Face, Flair, Tensorflow.

Below, the methodology followed by each team that published a paper addressing task 1 and/or task 2 is described. Teams are ranked based on the score achieved for the first MEDDOPROF task (i.e., NER) in descending order. Table 3.1 shows a summary of the result metrics obtained by the different teams.

3.4.1.1 NLNDE team

Neither Language Nor Domain Experts (NLNDE) team [111] (ranking in the tasks NER: 1st, CLASS: 1st), GitHub[2], employed XLM-RoBERTa (XLM-R) transformer model [112]. Briefly, XLM-R is a transformer-based multilingual MLM, pre-trained on one hundred languages (2% are Spanish documents) using CommonCrawl data, that has outperformed other multilingual models such as Multilingual BERT (mBERT). The NER task was addressed by the NLNDE team as a sequence labelling problem. The approach followed by the team consists of three phases:

- Further pre-training of XLM-R model with Spanish documents. According to the authors, adding domain knowledge in non-standard domains results in higher performance. This resulted in three models:

[2] https://github.com/boschresearch/nlnde-meddoprof

(i) Standard XLM-R model: original model as presented in [112].

(ii) Spanish XLM-R model with additional training with a medium-sized and general domain corpus.

(iii) Spanish clinical clinical XLM-R model with additional training with a small size clinical corpus.

- Transfer learning between the NER and the CLASS subtasks. As both tasks are related, the authors hypothesised that taking advantage of the knowledge of one of the tasks would benefit the other.
- Use of different data split strategies to train the sequence tagger:

(a) Training with all available data and use of the training loss stop criterion.

(b) Train-validation split based on document similarity using clustering techniques. The documents were clustered into five splits of the same size using k-means and Principal Component Analysis (PCA). Therefore, five models were trained and ensembled using a majority voting approach.

The pre-processing consists of tokenization at the subtoken level using XLM-R subword tokenizer and sentence segmentation, spaCy. The model architecture consisted of one of the XLM-R models plus a Conditional Random Fields (CRF) layer. The decision to use CRF was motivated by the need to address multiword annotations (usually presented in occupational data and to prevent inconsistencies in the labels). The sentences were split to have a maximum length of 300 subtokens and cross-sentence information was taken into account by considering 100 subtokens to the left and to the right. BIOSE schema was applied rather than BIO.

Finally, up to 43 models were developed. The best-performing model in the NER task was the XLM-R model with further training using the general domain corpus and applying strategic datasplits.

A year later, the authors from NLNDE team published CLIN-X [89] achieving an 81.68 F1-score on task 1 and 80.54 on task 2.

3.4.1.2 MUCIC team

MUCIC team [113] (ranking in the tasks NER: 2$^{\text{nd}}$, CLASS: 2$^{\text{nd}}$), GitHub[3], proposed two models based on BERT embeddings. The main component of both models was BETO [114], a Spanish BERT language model trained on a corpus comparable in size to the one used for training BERT:

(i) BETO cased.

(ii) Flair framework: Flair [115], provides a framework for using various embeddings and language models. MUCIC team used a Spanish model and fine-tuned using a Bi-LSTM based sequence tagger.

The model that took advantage of Flair-BERT embeddings was the one that achieved the highest F1-score. PyTorch was employed for both models.

3.4.1.3 SINAI team

SINAI team [116] proposed three models (ranking in the tasks NER: 4$^{\text{th}}$, CLASS: 5$^{\text{th}}$), also based on BETO. The solution presented by this team encompassed both multiclass (1 model) and binary classification (2 models) approaches. For the last one, authors masked all classes under the same label with the purpose of discriminating between entities and non-entities tokens. In addition, for binary classification, further training was performed using data from the ProfNER shared task.

[3]https://github.com/fazlfrs/MUCIC-MEDDOPROF

spaCy was used for normalisation, to note: lowercase conversion and removal of accented and special characters. The best-performing model was the multiclass one. Moreover, they implemented an auto-evaluation software that highlighted the discrepancies between their predictions and the golden test, this is, an error analysis. With this software, they were able to study the discrepancies in the solution implemented. Finally, they proposed a Bi-LSTM model together with CRF for future lines.

3.4.1.4 Vicomtech team

Vicomtech team [117], GitHub[4] (ranking in the tasks NER: 5th, CLASS: 6th), treated the MEDDOPROF shared task as a whole, proposing a multi-task joint model that tries to solve all three tasks at once. To this end, the authors proposed two BERT models: BETO introduced earlier, and IXAmBERT [118] (i.e., a multilingual language pre-trained for English, Spanish and Basque using Wikipedia web pages of the three languages as a corpus), although they finally used the first one after validation in the development set. The team concluded that hyperparameter settings seem to have a large influence on the performance of the systems proposed, so more experimentation was encouraged.

3.4.1.5 TALP team

TALP team [119] (ranking in the tasks NER: 8th, CLASS: 8th), prioritised the issue of data imbalance and the training complexity. With this in mind, the team proposed three models, all based on DistilBERT [87], a smaller, lighter and faster version of BERT that greatly reduces the training time. The way followed by the team to handle unbalanced was up-sampling low-prevalence occupations classes (e.g., *Activity*) by replacing other entities and adding additional context. To ensure that the new context made sense, a general-purpose BERT model was used to discard unlikely examples, computing a likelihood score to rank synthetic examples.

A notable contribution of this paper was to address task 1 and task 2 as a single joint task. Therefore, two alternatives were proposed:

- Single output with the cross-concatenation of the occupation classes and family classes as the set of labels. This approach was discarded due to the degradation of the F1-score.
- Two independent outputs for each task

Unlike other proposed systems, authors used In-Out (IO) encoding. Besides DistilBERT, the model architecture contained a Bi-LSTM layer at the top of the transformer layer and an independent time-distributed fully connected layer. The authors also experimented with the weights of the DistilBERT transformer. First, the weights were initialised thanks to a pre-trained general-purpose multi-lingual model, distilbert-base-multilingual-cased. However, the authors explored freezing some layers during training.

Due to computational limitations, the authors split the documents into overlapping sequences of 128 tokens. Therefore, they were able to experiment with the balance of positive (i.e., sequences that contain an entity) and negative sequences, discussing the trade-off between precision and recall depending on the proportion of positive and negative sequences. Finally, the three proposed models were:

(i) DistilBERT, full weights fine-tuning, and no data augmentation

(ii) DistilBERT, full weights fine-tuning, with data augmentation

(iii) DistilBERT, no weights fine-tuning, and data augmentation

Whereas data augmentation balanced precision and recall scores, the best F1 results in the test set, were obtained with the model (i), this is, without data augmentation.

[4]https://github.com/Vicomtech

3.4.1.6 EdIE team

EdIE team [13] (ranking in the tasks NER: 11th, CLASS: 10th), GitHub[5], proposed different BETO systems for the different subtasks. A thorough analysis of the corpus was performed by the team, highlighting issues such as overlapping annotations. A pre-processing step was carried out comprising the following actions: conversion to lowercase, handling of special characters, and tokenization using spaCy.

Moreover, the EdIE team also addressed the under-representation issue and implemented an undersampling technique for filtering documents without a positive tag. As the SINAI team [116] did, EdIE also used ProfNER corpus to have a greater representation of professions in the training and validation sets. Summarising the different approaches applied by the team, the following models were proposed:

(i) BETO considering all training data

(ii) BETO applying undersampling techniques

(iii) BETO considering all training data and ProfNER corpus

(iv) BETO applying undersampling techniques and ProfNER corpus

Model (ii) achieved the best results for task 1, whereas model (i) achieved the best results for task 2. Finally, the EdIE team suggested the employment of an occupation dictionary to further improve the results.

Table 3.1: Summary of the main ideas proposed by the participating teams

Team	Architecture	Contributions	Findings / best model	Future work
NLNDE	XLM-RoBERTa + CRF	1. Further pre-training with general domain and clinical corpus 2. Transfer-learning between tasks 1 and 2 3. Strategic datasplits based on PCA BIOSE encoding	Strategic datasplits + further training using the general domain corpus	Exploration of different clinical corpora
MUCIC	1. BETO cased 2. Flair embeddings	Flair framework	Flair embeddings	LUKE
SINAI	BETO	Multiclass and binary classification approaches Further training using ProfNER	Multiclass BETO without further training	Bi-LSTM + CRF
TALP	DistilBERT	Single-joint task Data augmentation IO encoding	Data augmentation balance precision and recall No data augmentation achieves best F1-score	-
Vicomtech	1. BETO 2. IXAmBERT	Multitask joint model	Multitask is feasible to solve all the sub-tasks	Choose better hyperparameters
KaushikAcharya	Linear-chain CRF + L-BFGS	Recurrent model	10% of the ground truth entities fell under partial match	LSTM for feature extraction
Jharkawat	1. BETO 2. Multilingual BERT cased	Multilingual approach	BERT tokenizer is inefficient for this task Eliminate sentences without tags	1. XLNet 2. Optimizer for memory efficiency
EdIE	BETO	Undersampling Further training using ProfNER BIOSE encoding	Undersampling techniques were useful only for Task 1	Occupation dictionary

Conditional Random Fields (CRF), Bidirectional Encoder Representations from Transformers (BERT), Bidirectional Long-short term memory (Bi-LSTM), Principal Component Analysis (PCA)

3.4.1.7 KaushikAcharya team

Conversely to the other proposed models, the KaushikAcharya team [120] (ranking in the tasks NER: 12th, CLASS: -), GitHub [6], presented a system based on linear chain CRF. Parameter estimation

[5]https://github.com/vsuarezpaniagua/EdIE-MEDDOPROF
[6]https://github.com/kaushikacharya/clinical_occupation_recognition

was done using an optimisation algorithm, and L1 and L2 regularisation techniques were applied. After performing an error analysis, the authors concluded that almost 10% of the ground truth entities fell under partial match. Moreover, the authors proposed Bi-LSTM models for improving feature extraction as a future work.

3.4.1.8 Jharkawat (IITKGP) team

Jharkawat team [121] (ranking in the tasks NER: 13th, CLASS: -), GitHub[7], trained two BERT models: BETO and Multilingual BERT (cased) and provide the results based on partial matches rather than using exact matches. The best-performing model was BETO, according to the team, probably due to multilingual BERT being trained on less Spanish data. The error analysis performed by this team highlighted that: i) BERT tokenizer is inefficient for the dataset provided by the competition, as the lexicon does not include terminology from the healthcare industry. ii) There are a large number of phrases that lack entity tags. As a solution, the team proposed to increase the dataset or eliminate the sentences with no tags. Finally, the XLNet architecture and the use of efficient adapters were two lines suggested by the authors for future research.

3.4.2 Conclusions and future work of MEDDOPROF works

From the methodology applied by the different teams, the following ideas/conclusions are extracted as candidates to boost the baseline model that will be proposed in Chapter 5:

- NLNDE: strategic datasplits, XLM-R models and further training with general domain Spanish documents could boost the performance of the models achieving good results in terms of F1-score. CLIN-X model [89], is also a promising approach.
- MUCIC: flair framework [115] applicability to this NER task is an approach to consider, in view of the results obtained. This team also proposed LUKE [122], as a model to consider for future work.
- SINAI: Bi-LSTM plus CRF models could be an alternative approach to implement based on its recommendations.
- EdIE: the employment of additional training data, an occupation dictionary, and undersampling techniques for maximising the number of sentences with positive entities could help in the classification task.

The task organisers provided an analysis of the difficulties encountered by the participating teams and the particularities of the MEDDOPROF corpus that could hinder the task. Some of them are listed below, as this knowledge could be helpful in developing this work proposal:

- Ambiguity: occupations that can act as a noun or as an adjective (e.g., *physician (noun)/ clinical (adjective)*).
- Similar linguistic constructions but different meanings: *trabaja en la construcción* is a multi-word occupation whereas *trabaja en su huerta* is an activity.
- Indirect mentions and abbreviations: the occupation is not explicitly stated but can be intuited from the context.
- Mention length / resolution (e.g., *profesora* / *profesora de pintura sobre vidrio y restauración de vidrieras*): an occupation can be annotated with different levels of resolution from general to specific.

Finally, in Table 3.2 the results of the participating teams can be seen.

[7] https://github.com/jharkawat/meddoprof_shared_task

Table 3.2: MEDDOPROF shared-task results. Table extracted from IberLEF 2021 - MEDDOPROF video

Team Name	NER			CLASS			NORM		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
EdIE-KnowLab	0.585	0.712	0.643	0.604	0.604	0.604	0.165	0.193	0.178
Fadi	0.802	0.678	0.735	0.761	0.644	0.698	0.682	0.541	0.603
Galiza	0.731	0.597	0.657	-	-	-	0.72	0.482	0.577
gbali	0.786	0.586	0.671	0.726	0.538	0.618	-	-	-
HULAT-UC3M	0.412	0.53	0.464	-	-	-	-	-	-
ICC	0.741	0.435	0.549	0.662	0.377	0.48	0.567	0.388	0.461
IITKGP	0.654	0.5	0.567	-	-	-	-	-	-
KaushikAcharya	0.807	0.524	0.635	-	-	-	**0.72**	0.467	0.566
MUCIC	0.813	**0.788**	0.8	0.77	0.75	0.764	-	-	-
NLNDE	**0.855**	0.783	**0.818**	**0.83**	**0.759**	**0.793**	-	-	-
SINAI	0.821	0.74	0.778	0.775	0.69	0.73	0.593	0.541	0.566
SMR-NLP	0.854	0.751	0.799	0.802	0.699	0.747	-	-	-
TALP	0.761	0.465	0.698	0.694	0.588	0.637	0.675	**0.572**	**0.619**
URJC-UNED Team	0.765	0.706	0.734	0.71	0.664	0.686	-	-	-
Vicomtech NLP-team	0.758	0.739	0.748	0.71	0.691	0.701	0.488	0. 474	0.481
Baseline	**0.465**	**0.508**	**0.486**	**0.391**	**0.377**	**0.384**	**0.502**	**0.533**	**0.517**

P: Precision, R: Recall, F1: F1-Score

3.5 Applications of transformers in Spanish clinical settings

Transformer-based AI models are proving to have great potential when applied to Spanish biomedical text (e.g., clinical cases and electronic health records). Nonetheless, their application has been largely limited to collaborative and shared evaluation campaigns (i.e., CLEF, IBERLEF), with relatively little focus on applied clinical research.

These models have been used in a wide variety of clinical settings. For instance, in [123], the authors investigated the applicability of three transformers (i.e., mBERT, BETO, XLM-RoBERTa) to automatic ICD-10 clinical coding (i.e., to assign a list of ICD-10-ES diagnostic and procedural codes to the text) achieving a new State-of-the-art performance, with an F1-score ranging from 0.52 to 0.86. Additional efforts on coding, have been made recently in [124].

Spanish automatic disease mention extraction has been addressed in DISTEMIST shared task [125]. The best-performing system used a RoBERTa implementation [126], and obtained a 0.79 F1-score.

The identification of negation and speculation qualifiers that could change the meaning of the clinical notes was addressed in [127]. In this research, BETO outperformed other DL architectures not based on transformers such as BiLSTM-CRF, 0.92 and 0.80 F1-score respectively.

Authors in [128] used BERT to detect pharmacological substances, compounds and proteins in PharmaCoNER recognition task [129], obtaining a F1-score ranging from 0.84 to 0.91. The same team, [130], also used BERT to detect tumour morphology mentions in the CANTEMIST shared task [131] obtaining a 0.87 F1-score.

The best scoring proposal in LivingNER shared task for the recognition of species, pathogens and food [132], was achieved with BETO, with an F1-score ranging from 0.93 to 0.95 [133].

Other scenarios in which transformers have shown outstanding performance were: i) the automatic correction of real-word errors in Spanish Clinical Texts [134], ii) machine translation of clinical texts from Basque to Spanish [135] where transformers showed a better performance than recurrent neural networks, iii) lung cancer information extraction [136] where transformers also performed better than Bi-LSTM models and iv) the identification of laterality, location, and findings from mammographic radiological reports with a 0.88 F1-score [137].

Chapter 4

Materials and Methods

In this chapter, the main dataset used in this work, MEDDOPROF, is formally presented and described, together with a manually annotated corpus exclusively for this Master's thesis, called More Occupation Data (MOD). The steps performed to build and annotate this corpus are explained. Finally, the libraries and the working environment employed for conducting the experiments are also formally introduced.

4.1 MEDDOPROF corpus description

MEDDOPROF is a public corpus consisting of 1,844 Spanish clinical case reports with annotations for occupations, working status, and activities. The clinical case reports came from more than 20 specialities. Table 4.1 shows the frequency of the notes depending on the medical speciality. The corpus is provided in *.zip* format and presents two different files per clinical case: a *.txt* file with the original note and a *.ann* file with the annotations. Both files are associated with the file-naming convention. The corpus is structured into three folders with different levels of annotation and labels:

- MEDDOPROF-NER: comprise .ann and .txt files with profession (OCUPACION), working status/employment status (SITUACION_LABORAL), and activity (ACTIVIDAD) annotations.
- MEDDOPROF-CLASS: comprise .ann and .txt files with patient (PACIENTE), family member (FAMILIAR), health professional (SANITARIO), and other (OTROS) annotations.
- ner-class-joint: comprise .ann and .txt files with both levels of annotation joint, this is, NER-CLASS (e.g., PATIENT-OCCUPATION).

A MEDDOPROF-NER annotation always has an attached annotation specifying the subject to which the occupation belongs, MEDDOPROF-CLASS, this is the same number of mentions are considered for task 1 and task 2. For example:

Sentence:	Paciente	trabajador	de	la	construcción	jubilado
MEDDOPROF-NER	O	B-PROFESION	I-PROFESION	I-PROFESION	I-PROFESION	B-SITUACION_LABORAL
MEDDOPROF-CLASS	O	B-PACIENTE	I-PACIENTE	I-PACIENTE	I-PACIENTE	B-PACIENTE

Finally, there is a *.tsv* file with the mapping of each mention in the corpus to the European Skills, Competencies, Qualifications and Occupations (ESCO)[1] and SNOMED CT[2] terminologies.

The annotations follow the BRAT standoff format. In this format, each line contains one annotation, and each annotation receives an ID that appears first on the line, separated from the rest of the annotation by a single-tab character. The rest of the structure varies by annotation type [138]. A detailed picture of the annotation schema can be seen in Figure 4.1.

[1] https://esco.ec.europa.eu

[2] https://www.snomed.org/

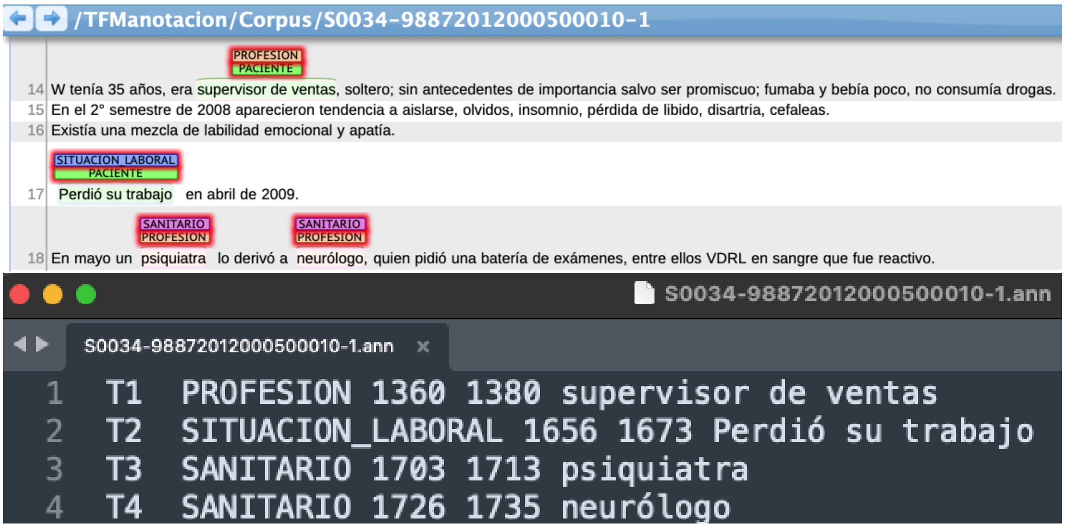

Figure 4.1: Sub-task 1 annotation schema. Source: https://temu.bsc.es/meddoprof/tracks/. Further details regarding the BRAT *standoff* schema can be seen in https://brat.nlplab.org/standoff.html

The corpus annotation guidelines are available online[3]. Briefly, a pre-annotation step was carried out using semi-supervised learning methods, then professional annotators checked the automatic pre-annotation. 500 case reports were annotated by two experts to develop and refine the annotation guidelines. A mean of 0.9 Inter-Annotator Agreement (IAA) was obtained after multiple annotation rounds. The corpus is divided into the train (1,500 case reports) and test (344 case reports) sets. It contains 1,291,186 tokens, 4,743 manual annotations, and 346 unique codes (i.e., 297 ESCO and 49 SNOMED-CT codes). More details of the corpus are shown in [37]. The number of documents, annotations, unique codes, sentences, and tokens per dataset is shown in Table 4.2. Finally, the distribution of the entities of task 1 (i.e., MEDDOPROF-NER) and task 2 (i.e., MEDDOPROF-CLASS) can be seen in Table 4.3. An inconsistency was found in the test subset, and four annotations for the same entity were detected in *caso_clinico_psiquiatria304.ann*, two with the same tag (i.e., family) and the other two with another tag (i.e., patient). Furthermore, duplicate notes ($n = 2$) were found in the training (caso_clinico_atencion_primaria161 - caso_clinico_atencion_primaria162) and the test sets (casos_clinicos_profesiones120 - casos_clinicos _profesiones193). The low number of annotations compared to the number of sentences shows a scenario in which negative sequences (i.e., with no entities) prevail. From this last table, data imbalance can be appreciated, where the *Activity* label from task 1 and the *Family* label from task 2 constitute the minority classes, accounting for the 3% and 5% of cases, respectively. In addition, in Table 4.4 the number of characters, tokens, and entities are shown. Finally, some annotated documents contain partial overlapping annotated entities, an allowed scenario, according to the annotation guidelines. The median and quartile 25 (Q1) and 75 (Q3) number of words per entity are 2 (1-4), in both the training and test sets.

Finally, minor discrepancies in the number of "tokens" and "entities" exist between Table 4.3 and Table 4.4 as the number of tokens may vary with the tokenization implementation used. All the statistics shown in these tables were calculated with the *Estadisticas.ipynb* script. As explained above, some of the figures obtained with this script slightly differ from the ones provided by the authors of Tables 4.3 and 4.4.

4.2 Additional training data: More Occupation Data corpus (MOD)

Other Spanish and English corpus to enrich the MEDDOPROF training set and/or reduce the number of negative sentences and the imbalance were considered. This was done following the tasks' organisers' advice: *"need to expand the annotated data to 2k documents"*. The actual number

[3] https://zenodo.org/record/4720833

Table 4.1: MEDDOPROF clinical notes specialities

Speciality **N (%)**	**total** **n = 1,844**	**train** **n = 1,500 (0.81)**	**test** **n = 344 (0.19)**
Psychiatry	560	484 (0.86)	76 (0.14)
Labour	233	81 (0.35)	152 (0.65)
Internal medicine	229	207 (0.9)	22 (0.1)
Oncology	194	175 (0.9)	19 (0.1)
Primary care	93	86 (0.92)	7 (0.08)
Dermatology	87	77 (0.89)	10 (0.11)
Infectology	65	58 (0.89)	7 (0.11)
Neurology	63	54 (0.86)	9 (0.14)
Other II	58	50 (0.86)	8 (0.14)
Emergency	35	34 (0.97)	1 (0.03)
Radiology	31	27 (0.87)	4 (0.13)
Otorhinolaryngology	28	26 (0.93)	2 (0.07)
Allergology	25	24 (0.96)	1 (0.04)
Odontology	24	22 (0.92)	2 (0.08)
Ophthalmology	24	22 (0.92)	2 (0.08)
COVID	20	19 (0.95)	1 (0.05)
Urology	20	16 (0.8)	4 (0.2)
Other I	19	16 (0.84)	3 (0.16)
Tropical medicine	18	15 (0.83)	3 (0.17)
Endocrinology	10	7 (0.7)	3 (0.3)
Rheumatology	8	0 (0)	8 (1)

Other I: includes all documents starting with SXXXX-. Other II: includes all documents starting with XXXXXXXX_ES

Table 4.2: Number or documents, annotations, unique codes, and sentences in the MEDDOPROF corpus. Table extracted from IberLEF 2021 - MEDDOPROF video

	Documents	Annotations	Unique Codes	Sentences	Tokens
Train	1,500	3,658	297	49,114	1,075,655
Test	344	1,085	167	9,513	215,531
Total	1,844	4,743	346	58,627	1,291,186

Table 4.3: Proportion of entities in the MEDDOPROF corpus. In parentheses, train and test proportions

	Patient	Family	Health Prof.	Other	Total
Profession	1,158 (876-282)	134 (105-29)	1,525 (1,231-294)	410 (316-94)	3,227 (68.04%) (2,528-699)
Empl. Status	1,047 (754-293)	119 (97-22)	0	203 (160-43)	1,369 (28.86%) (1,011-358)
Activity	122 (105-17)	7 (5-2)	0	18 (9-9)	147 (3.10%) (119-28)
Total	2,327 (49.06%) (1,735-592)	260 (5.5%) (207-53)	1,525 (32.14%)	631 (13.29%) (485-146)	4,743 (3,658-1,085)

Table 4.4: Descriptive statistics of MEDDOPROF corpus: characters, tokens, and entities. Table extracted from [116]

Metric	Train			Test		
	Average	Min-Max	Total	Average	Min-Max	Total
Number of characters in document	4,159.72	184 - 27,529	6,239,588	3,606.29	228 - 23,446	1,240,562
Number of tokens in document	743.28	29 - 4,807	1,114,919	647.51	37 - 4,376	222,744
Number of entities in document*	7.44	1 - 86	9,217	9.05	1 - 79	2,786

*The number of entities considers both beginning (B) and inside (I) tags

of annotated documents was 1,844.

Candidates must meet one of the following conditions: i) the corpus is already annotated with occupation mentions, or ii) the corpus belongs to the clinical setting.

4.2.1 Spanish corpora

Hereafter, some potential Spanish corpora for enriching the training dataset are proposed:

- MEDDOCAN-SPACCC (GitHub): MEDDOCAN (Medical Document Anonymization) - SPACCC (Spanish Clinical Case Corpus) [139] contains one thousand Spanish clinical cases (train = 500, validation= 250, test = 250), approximately 33 thousand sentences and is annotated with 29 entity types (e.g., dates, email, country, name, age) . Profession is one of them, with 37 annotated mentions. This corpus was annotated with the aim of anonymising medical documentation and was also distributed using the BRAT standoff format.

  As this corpus was already annotated for professions, a script, *TransformacionAnotacion-MEDDOCAN.ipynb*, was used to remove the rest of the annotation entities from the *.ann* files. 35 clinical notes with 37 occupation mentions were found.

- ProfNER (GitHub): ProfNER [38] comprises social media data, more specifically, of 10,000 tweets (train = 6,000, validation = 2,000, test = 2,000 / background = 13,500) related to the COVID-19 pandemic in Spanish, annotated with mentions of professions and occupations. The files were provided in BRAT standoff format. Note that the test/background set is not annotated.

- NUBes-IULA (GitHub): NUBes corpus [140] is a collection of 608 anonymised Spanish clinical notes, containing 29,682 sentences annotated for negation and uncertainty. The median, Q1-Q3 number of tokens per corpus sentence is 14 (9-23). Sentences of this corpus are shuffled. On its behalf, IULA corpus [141] is a Spanish corpus annotated for negation, containing 3,194 sentences (recently also annotated for abbreviations [142]). The corpus is provided in seven different files, each containing around 470 sentences. The median, Q1-Q3 number of tokens per sentence is 10 (6-14) [127]. Sentences of this corpus are shuffled to avoid traceability and separated using the "-" character. Both corpora are public and distributed in BRAT standoff format, but no occupation information can be found in any of those.

- Other Spanish clinical notes corpus such as BARR2, for abbreviation recognition [143]; CANTEMIST, with oncology clinical annotations [144]; CodiEsp, with Spanish clinical cases, [145]; LivingNER with species, pathogens and food mentions in clinical notes [146]; PharmaCoNER, with pharmacological substances, compounds and proteins mentions [147] and DisTEMIST with disease annotations were also considered [148].

  Finally, other Spanish corpora were immediately discarded, such as CARES with radiological reports [149] or The Chilean Waiting List Corpus a corpus with referrals from the waiting list in Chilean public hospitals [150], as they were out of the scope of this work (i.e., the first one is limited to radiological data and the nature and the structure of the second one differs from our objective). Not easily accessible corpus such as IxaMed-GS, annotated with adverse drug reactions [151] or UHU-HUVR [152], annotated with negation, were excluded. Crawled corpus such as CoWeSe [153] or Spanish ADR corpus [154] were not included. The main exclusion reasons for CoWeSe were: data extracted from URLs and not purely based on clinical cases and plain text corpus (i.e., just one file). The main exclusion reason for the Spanish ADR corpus was its nature: comments extracted from social media annotated with drugs and adverse events.

As many of the Spanish corpus presented above came from the same source, Text Mining Unit (TEMU) at Barcelona Supercomputing Center, and some clinical notes are present in several corpora (e.g., BARR2 and PharmaCoNER share most of the notes), an analysis of the notes selected for annotation was carried out to ensure (filename and note content) that none of those, which would be used to improve training, were also present in the original training or test sets. Otherwise, data leakage could occur.

4.2.2 English corpus

English corpora that could be used to enrich the training dataset and use a cross-lingual / multilingual approach are shown:

- SHAC: SHAC corpus [94] is an English corpus with annotations of SDOH, such as *employment* and *employment status* (i.e., tobacco, statustime, alcohol, amount, frequency, drug, type, livingstatus, typeliving, employment, statusemploy, statustimeval, typelivingval, statusemployval, method, duration, history). The number of notes in the train set is 1315, whereas the number of notes in the development set is 188. This corpus was distributed in a n2c2 task [96], and the files also followed a BRAT format. As this corpus is not publicly available, it was not used. Data access to *n2c2 NLP Research Data Sets* was requested and granted on January, 3$^{\text{rd}}$ 2023. To fulfil the *NLP Data Use Agreement*, data could only be used for evaluation purposes.

- MIMIC-III [95]: MIMIC-III is a relational, large, de-identified and publicly available database consisting of 26 tables, including clinical notes (n = 112000), with an average of 709 tokens, from more than 40.000 patients admitted to critical care units (and some neonates data). Free text data include provider progress notes, hospital discharge summaries, and free text reports of electrocardiogram and imaging studies. Data access was formally requested, after completing *Data or Specimens Only Research* course, and a PhysioNet[155] account was created as a requirement. Data access was granted on January, 10$^{\text{th}}$, 2023.

- CodiEsp, LivingNER, ProfNER contained both Spanish and English clinical annotations. For a brief description, see the previous section.

4.2.3 Data selection

To automatically select notes to annotate from the corpus introduced in Section 4.2, a rule-based algorithm based on regular expressions and exact string matching, and a Spanish gazetteer of occupation mentions (provided in the ProfNER corpus) was used to identify potential notes with occupation mentions. The gazetteer contained more than 25,250 occupations mentions. Briefly, a gazetteer is a list of entities that acts like a look-up dictionary. The gazetteer can be used to identify the entities by matching them in the text. It was processed in three different ways to maximise the number of matches:

(i) Select the first four words and remove duplicates (n = 21,077)

(ii) Select the first word and remove duplicates (n = 4,305)

Table 4.5: Corpus considered for enriching the training set

Corpus name	**Language**	**Description**	**Occupation annotations**	**Size**	**Accesibility**
BARR2 [143]	Spanish	Biomedical abbreviations (Clinical notes)	No	Train = 318 Dev = 146 Test = 220 Background = 2,879	Public
CANTEMIST [144]	Spanish	Cancer annotations in clinical records	No	Train = 501 Dev = 500 Test = 300 Background = 4,932	Public
CodiEsp [145]	Spanish & English	Diagnoses and procedures annotations (Clinical notes)	No	Train = 500 Dev = 500 Test = 500 Background = 2,751	Public
IULA [141]	Spanish	Negation in clinical records	No	7 clinical notes 3,194 sentences	Public
LivingNER [146]	Spanish & English	Animals, plants, and microorganisms (Clinical notes)	No	Train = 1,000 Dev = 500 Test = 500 Background = 12,972	Public
MEDDOCAN [139]	Spanish	Medical documents anonymization (Clinical notes)	Yes	Train = 500 Dev = 250 Test = 250	Public
NUBes [140]	Spanish	Negation and uncertainty in biomedical texts (Clinical notes)	No	608 clinical notes 29,682 sentences	Public
PhamaCoNER [147]	Spanish	Pharmacological substances, compounds and proteins (Clinical notes)	No	Train = 500 Dev = 250 Test = 250 Background = 2,751	Public
ProfNER [38]	Spanish	Professions & occupations in health-related social media (Tweets)	Yes	Train = 6,000 Dev = 2,000 Test = 2,000 Background = 25,000	Public
MIMIC-III [95]	English	EHR with 26 tables	No	+100k	Restricted
SHAC n2c2 [94]	English	Social determinants of health (Clinical notes)	Yes	Train = 1315 Dev = 188	Restricted

(iii) Select the first word after stemming to avoid string match failure due to gender (i.e., male/female) or number (i.e., singular/plural) differences (n = 3,181)

The case-sensitive match was deactivated to avoid discarding notes due to the caps' appearance. In addition, some words were used as stopwords after manual assessment to reduce false positive cases. Each word in the gazetteer was matched with each clinical note. If a match occurs, the name of the candidate note to annotate is compared with the name of all MEDDOPROF test notes and discarded if there is a match (this works for the corpus annotated by the TEMU team). Otherwise, the clinical note filename was stored, and the note was identified as a final candidate for annotation.

The approach that maximised the number of matches, that is, the one that uses stemming (iii) was finally chosen. In addition to the gazetteer, a rule-based match pattern was used. All cases of clinical notes containing any of the following strings were selected: *trabaj*/ocupacion/profesion*, independently of the gazetteer results. Using rules for identifying occupations promotes false positive appearances. For example, notes without occupations that contained the following strings were identified: *"trabajo respiratorio", "trabajo de parto"*.

The initial number of notes in each corpus was: nBARR2 = 3,563, nCANTEMIST = 6,233, nCodiEsp = 3,751, nIULA = 7, nLivingNER = 14,972, nMEDDOCAN = 1,000, nNUBEs = 608, nPharmaCoNER = 3,751, nProfNER =35,000. From this point on, different inclusion and exclusion criteria were applied to select the notes to annotate. A diagram of this whole process can be found in Figure 4.2. The steps taken were:

1. Corpora selection: ProfNER corpus was not included, as its nature and statistics (i.e., tweets rather than clinical notes) differ from the rest (special characters, not clinical language, and so on). Corpora with notes shuffled were also excluded as it is difficult to establish to whom the occupation belongs. For instance, the following sentence can be found between other non-related sentences *Trabaja de arquitecto*. No information regarding to whom the occupation belongs is accessible. Therefore, IULA corpus was also discarded since there were only seven files that contained multiple and shuffled clinical notes. In addition, NUBes corpus was also discarded, as each file contains shuffled sentences from multiple clinical notes (although all the sentences in a file corresponding to the same speciality).

2. Gazetteer and rule-based algorithm: the number of notes identified by the gazetteer and the rule-based algorithm was: nBARR2 = 2-184, nCANTEMIST = 2-319, nCodiEsp = 2-184, nLivingNER = 8-1,102, nMEDDOCAN = 0-45, nPharmaCoNER = 2-184. Some of the notes were simultaneously identified by the gazetteer and the rule-based algorithm. As can be appreciated, the gazetteer did not perform as expected and the number of notes retrieved was low. However, the rule-based algorithm was able to detect a non-negligible number of notes. Duplicate notes belonging to the same corpus (e.g., notes belonging to the test set and the background set at the same time) were identified in the LivingNER and BARR2 corpus, and removed.

3. Notes in MEDDOPROF: Notes from the remaining six corpus that were also present in the MEDDOPROF train or test sets were discarded (nBARR2 = 151, nCANTEMIST = 226, nCodiEsp = 157, nLivingNER = 477, nMEDDOCAN = 29, nPharmaCoNER = 157; nTotal = 1,197).

4. Duplicate notes: From the selected notes, those present in more than one corpus (i.e., had the same filename), n = 715, were excluded. Later on, an analysis of the notes content was performed (Consisting of lowercase conversion, special character removal, stopwords deletion, and stemming) to discard possible duplicates, with a different filename. In fact, n = 5, were identified using the *duplicadosNotas.ipynb* script (for further details, see Appendix A.2). These notes differed in the indentation (e.g., different number of carriage returns) level. For this reason, they were not identified at the beginning of the pre-processing pipeline. From each pair of duplicate notes, the one starting with "S" was kept, and the other was removed.

5. Duplicate notes based on TF-IDF score: duplicate notes that slightly differ (i.e., one contains a header and the other does not, but the note is essentially the same) exist. To detect and exclude these cases, a similarity matrix was built using a TF-IDF approach. Then, the pair of notes with a TF-IDF value greater than a cutoff of 0.35 was chosen for review. This was done in two steps: first, the similarity was measured only in the candidate notes to annotate. Then, the similarity was measured between the candidate notes to annotate and the train and test sets from MEDDOPROF. Briefly, with TF-IDF a document similarity analysis is performed.

6. Manual review: three duplicate notes not identified by the previous steps were identified and removed. Duplicate notes that were excluded, as well as the exclusion reason, are shown in Table A.2.

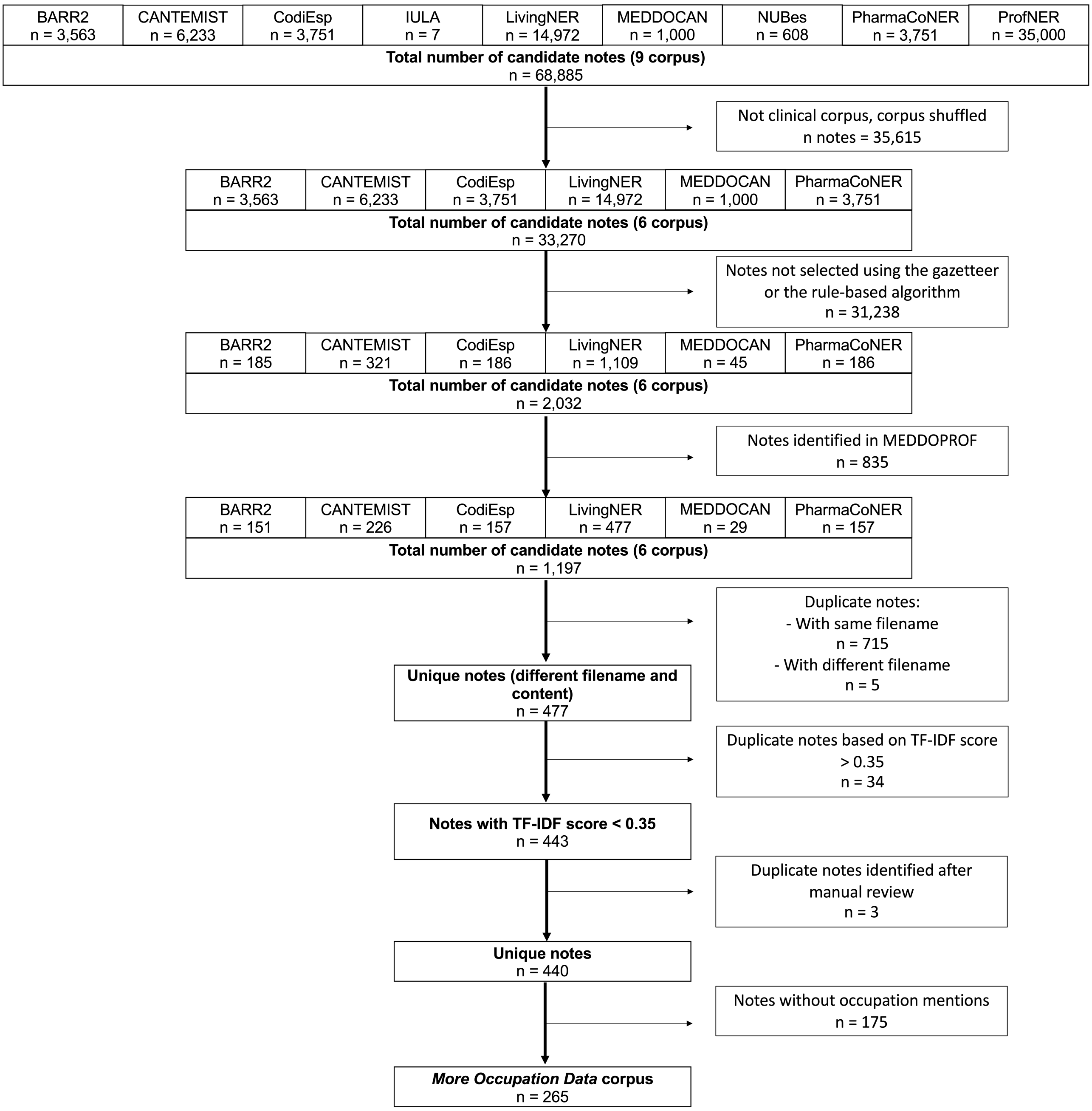

Figure 4.2: MOD corpus exclusion and inclusion criteria

After applying the exclusion and inclusion criteria, 265 notes remained and opted for annotation. The set of these notes was called More Occupation Data (MOD) corpus, and the annotation process was conducted. Only 265 notes out of 440 contained occupation mentions and were annotated. The percentage of notes incorrectly identified by the rule-based algorithm and the gazetteer (i.e., false positives) after removing duplicates was 40%. The entire data selection process was implemented using *NotasArevisar.ipynb*, *duplicadosNotas.ipynb*, and *ExtraccionNotas.ipynb* scripts, accessible via GitHub (see Appendix A for further details).

4.2.4 Annotation and BRAT tool

BRAT was used to annotate clinical notes from the previous corpus. MEDDOPROF guidelines [156] were considered to annotate the different entities. Accordingly, for the first task (i.e., MEDDOPROF-NER) the possible tags were:

- Profession (OCUPACION): occupations that provide a person with an income or livelihood, including conventional professions, civil servants, public employees, new professions, and illegal professions. *'Ex'* and *'Co* prefixes are considered part of the profession.
- Working status (SITUACION_FUNCIONAL): including homemaker; retired; unemployed; unpaid caregiver; student, PhD student, apprentice, competitive examinations student; under temporary employment regulation; self-employed; on maternity/paternity leave; slave; prisoner, homeless, pauper; worker; other unspecified professional; refugee; hourly, full-time, part-time job; military service; military veteran; and co-worker or colleague.
- Activities (ACTIVIDAD): non-remunerated professions such as non-professional athlete/entertainer; unpaid community positions; activist; volunteer; guru or gamer.

For the second task (e.g., MEDDOPROF-CLASS):

- Patient (PACIENTE): main actor of the clinical note.
- Familiar (FAMILIAR): family member related to the patient.
- Health professional (SANITARIO): health-related professional who interacts with the patient, namely primary and secondary doctors, nurses, and assistant nurses.
- Other (OTROS): other people mention not captured in any of the categories above.

Seventy-one rules were described in the annotation guidelines provided by the task organisers. Only cases that were clear enough and in agreement with the guidelines were considered. More details can be found in Appendix A.1.

An interesting finding of this annotation process was to identify some clinical specialities that tend to write the patient's occupations, such as tropical medicine, while in others, occupation mentions are not that relevant (e.g., neurology, odontology emergency). Furthermore, this corpus was born to identify occupations at higher risk in the COVID-19 pandemic outbreak; however, the prevalence of this information in the manually annotated notes pertaining to this speciality was low.

The manual annotation process is a difficult, time-consuming, and exhaustive labour that has been identified as the bottleneck of many NLP tasks [157]. Although all manual annotations were reviewed within days after the first annotation, this step is prone to errors. The review was carried out to minimise their impact. An active learning approach was considered but finally not implemented; see Section 2.2.4.

The selection of brat rapid annotation tool (BRAT)[138] as the annotation tool was based on the following factors:

- The TEMU-BSC corpus is already annotated with BRAT, following the standoff format
- It is widely accepted by the research community and scripts for converting standoff format to BIO are easily accessible.

A comparison of other annotation tools has been made in [157] and [158]. Finally, the deployment of BRAT is addressed in Appendix A.1.

4.2.5 MOD corpus descriptive statistics

To replicate the statistics shown in Tables 4.2, 4.3 and 4.4, they are also calculated for the MOD corpus and presented in Tables 4.6, 4.7, 4.8.

As the average length of sentences varied from corpus, and the MEDDOPROF corpus contained a non-negligible number of negative sentences, two options were considered: training the algorithm only with positive sentences (i.e., with entities) or including all sentences (this could worsen the imbalance scenario). Finally, all the sentences belonging to a clinical case with at least one occupation-related entity were considered.

Table 4.6: Number or documents, annotations, and sentences in MOD corpus

Corpus	Documents	Annotations	Sentences	Tokens (NeuroNER)
MOD	265	639	9,746	223,891

Table 4.7: Proportion of entities in MOD corpus

	Patient	Family	Health Prof.	Other	Total
Profession	186	9	185	41	421 (65.88%)
Empl. Status	133	32	0	13	178 (27.86%)
Activity	40	0	0	0	40 (6.26%)
Total	359 (56.18%)	41 (6.42%)	185 (28.95%)	54 (8.45%)	639

Table 4.8: Descriptive statistics of MOD corpus: characters, tokens and entities

Metric	MOD corpus		
	Average	Min-Max	Total
Number of characters in document	4,879.29	567 - 25,706	1,293,011
Number of tokens in document (spaCy)	861.61	95 - 4,720	228,326
Number of entities in document*	6.14	1 - 75	1,626

*The number of entities considers both beginning (B) and inside (I) tags

4.3 Hospital Clínico San Carlos Musculoskeletal Cohort

MediLog, deployed in April 2007, was the first departmental EHR used in the HCSC Rheumatology Service, in operation until the end of 2018. It was designed to assist physicians in the patient healthcare provision while facilitating secondary uses of data, including research. More details of the cohort are provided in [30].

Clinical narratives from MediLog are used to evaluate the performance of the best-performing system trained in objectives 1 and 2. Only each patient's first visit is retrieved. This maximises the probability of finding occupation mentions. After data cleaning, described in [30], 35,586 first visits from 2007 to 2017 are considered. A histogram with the age of patients at first visit is displayed in Figure 4.3. It is important to take into account the aforementioned figure, given that the average retirement age of the Spanish population is 65 years and the average age of the patients in this cohort is high.

In addition, an example of a fictitious clinical note from MediLog is shown in Figure 4.4. The free-text content is delimited by tags. A script, not publicly available to comply with data protection, is developed to extract this content from the rest of the clinical note.

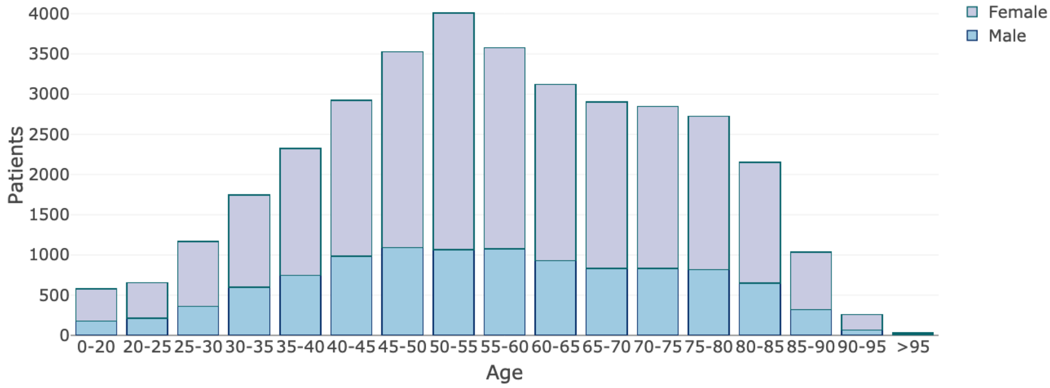

Figure 4.3: Patients - number of visits

```
999999  1980-01-01  EVOLUCION DE LA ENFERMEDAD Y EXPLORACION desde hace 8 meses dolor y mayor
discapacidad/impotencia funcional en hombro izquierdo tiene rotura ha empeorado de la rodilla y
posiblemente tenga derrame pongo analgésico   COMORBILIDAD  hipertensión ansiedad MEDICACION
CONCOMITANTE  brainal masdil orfidal eutirox zocor seropram sinvastatina codiovan idaptan
ALERGIAS A MEDICACION  no alergias  EFECTOS ADVERSOS    DIAGNOSTICOS  Sindrome del manguito
rotatorio Poliartrosis TIR  SITUACION FUNCIONAL  Situacion Laboral:AMA DE CASA Distress: C.
MODERADO Discapacidad2. Ligera discapacidad social Rosser: 0,973    SEGUIMIENTO     TRATAMIENTO
APLICADO  Diazepan 5 comprimidos 5 mg 1, cada 24 h antes de acostarse. Voltaren retard comp 75mg
1, cada 12h con las comidas    PETICIONES PRUEBAS  Laboratorio Central COMENTARIO  rev en 6
meses
```

Figure 4.4: Example of a clinical note from MediLog. The data presented has been created for illustrative purposes

A stratified selection of 2,000 clinical notes, organized by year, is randomly chosen for annotation. The distribution is shown in Table 4.9.

Table 4.9: Number of selected notes by year

2007	348	**2010**	100	**2013**	115	**2016**	101
2008	381	**2011**	101	**2014**	86	**2017**	313
2009	228	**2012**	100	**2015**	127		

The characteristics of this dataset are shown in Tables 4.10, 4.11, and 4.12.

Table 4.10: Number of documents, annotations, and sentences in the selected HCSC MediLog notes

Corpus	Documents	Annotations	Sentences	Tokens
HCSC MediLog	2,000	756	15,306	202,173

Table 4.11: Proportion of entities in HCSC selected notes

	Patient	Family	Health Prof.	Other	Total
Profession	148	6	518	2	674 (89.15%)
Empl.Status	54	0	0	0	54 (7.14%)
Activity	28	0	0	0	28 (3.7%)
Total	230 (30.42%)	6 (<1%)	518 (68.51%)	2 (<1%)	756

The limited occurrence of references to employment/working status entities within the free text notes can be explained by the presence of a four-category variable that encapsulates this information (i.e., active, student, retired and housekeeper) in a structured manner.

New cases not previously seen in the MEDDOPROF or MOD corpus arise in the HCSC notes:

- New abbreviations such as, *mp* (i.e., médico de primaria), *mdc/mdec* (i.e., médico de cabezera). According to rule P1 of the MEDDOPROF guidelines, these abbreviations were not annotated.

Table 4.12: Descriptive statistics of the selected HCSC MediLog notes: characters, tokens and entities

Metric	HCSC selected notes		
	Average	Min-Max	Total
Number of characters in document	558.17	10 - 5,299	1,116,341
Number of tokens in document (spaCy)	101.09	1 - 1,008	202,173
Number of entities in document*	0.6	1 - 13	1,209

*The number of entities considers both beginning (B) and inside (I) tags

- Typos in the entities, as *psicolologa* (i.e., instead of *psicóloga*).
- Words/entities not separated by spaces such as, *camareronocturno*.
- Mixed entity types (family member/healthcare professional) such as, *El paciente es sobrino del Dr XXX, que lo refiere para evaluación.*
- In some mentions, the *working status* and the *activity* entities appeared together: *estuvo yendo a trabajar como camarero y a natación durante 2 años.* In these cases, the MEDDOPROF rule P8 applies.

These notes also feature the following particularities:

1. Shorter and simpler notes.
2. Abundant spelling mistakes and typos.
3. Fewer occupationally related entities.
4. Abundant references to health professionals.
5. Highly repetitive entities. For instance, MAP (i.e., *médico de atención primaria*) is present in a large number of notes.

HCSC Ethics Review Board approval for retrospective studies and waiver of informed consent was obtained for the use of deidentified clinical records (23/340-E).

4.4 Tools and resources

Python 3.8.16 is used to carry out the experiments. The reasons behind this decision are: (i) most of the pre-trained models have been trained with Python, (ii) there are a large number of NLP libraries written in Python, (iii) it is supported by the community and extended documentation is available, and (iv) it is open source. Together with this programming language, several libraries have been proposed to conduct the experiments, as described below.

4.4.1 Libraries and frameworks

Data manipulation, algorithms and models main libraries used in this work encompass the following ones:

- Pandas (≥1.3.5): Data manipulation library. According to the official documentation:

pandas is a fast, powerful, flexible and easy-to-use open source data analysis and manipulation tool, built on top of the Python programming language. pandas provides high-level data structures and functions designed to make working with structured or tabular data intuitive and flexible [159].

- HuggingFace's Transformers (≥4.25.0): Transformers models library for PyTorch, TensorFlow, and JAX. According to the official website:

  > *Transformers provides APIs and tools to easily download and train state-of-the-art pretrained models. Using pre-trained models can reduce your compute costs, carbon footprint, and save you the time and resources required to train a model from scratch. These models support common tasks in different modalities, such as: natural language processing, computer vision, audio, multimodal [160].*

- Scikit-learn (≥1.2): Machine learning algorithms library and evaluation metrics. According to the developers:

  > *Scikit-learn is a Python module integrating a wide range of state-of-the-art machine learning algorithms for medium-scale supervised and unsupervised problems. This package focuses on bringing machine learning to non-specialists using a general-purpose high-level language. Emphasis is put on ease of use, performance, documentation, and API consistency [161].*

NLP dedicated libraries:

- Natural Language Toolkit (NLTK) (≥3.7): according to the official NLTK website:

  > *NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum [57].*

- spaCy (≥3.4): according to the official website:

  > *spaCy is a library for advanced natural language processing in Python and Cython. It's built on the very latest research, and was designed from day one to be used in real products. spaCy comes with pretrained pipelines and currently supports tokenization and training for 70+ languages. It features state-of-the-art speed and neural network models for tagging, parsing, named entity recognition, text classification and more, multi-task learning with pretrained transformers like BERT, as well as a production-ready training system and easy model packaging, deployment and workflow management.*

Deep-learning frameworks:

- PyTorch (≥1.13): according to the developers:

  > *PyTorch is a machine learning library that shows that speed and usability are compatible: it provides an imperative and Pythonic programming style that supports code as a model, makes debugging easy and is consistent with other popular scientific computing libraries, while remaining efficient and supporting hardware accelerators such as GPUs [162].*

- Keras Tensorflow (≥2.11): according to the official website:

  > *Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow [163]. It was developed with a focus on enabling fast experimentation. "Being able to go from idea to result as fast as possible is key to doing good research" [164].*

Evaluation libraries considered:

- seqeval (≥0.0.10): according to the developers:

  > *seqeval is a Python framework for sequence labeling evaluation. seqeval can evaluate the performance of chunking tasks such as named-entity recognition, part-of-speech tagging, semantic role labeling and so on [165].*

- nereval (≥ 0.2.5): according to the developers:

  > *Evaluation script for named entity recognition (NER) systems based on entity-level F1 score. It evaluates an NER system according to two axes: whether it is able to assign the right type to an entity, and whether it finds the exact entity boundaries.*

  This library was not finally used as the input format is a .json instead of .ann file.

- nervaluate (≥ 0.1.8): according to the developers:

  > *nervaluate is a python module for evaluating Named Entity Recognition (NER) models as defined in the SemEval 2013 - 9.1 task. The evaluation metrics output by nervaluate go beyond a simple token/tag based schema, and consider diferent scenarios based on wether all the tokens that belong to a named entity were classified or not, and also whether the correct entity type was assigned.*

From a list of 144 transformers models[4], all were supported on PyTorch, 61 were supported on Tensorflow and 30 on Flax. As the adoption of PyTorch is higher than in other frameworks, PyTorch was finally chosen as the DL framework for carrying out the experiments. Other authors have also highlighted additional reasons for choosing PyTorch over other frameworks: flexibility, dynamicity and easier to prototype and debug [51].

4.4.2 Working environment

4.4.2.1 Training tools

Since model training can be computationally expensive, the use of GPU-backed Jupyter notebooks was planned. There are different cloud providers that facilitate free computing notebook resources, including GPUs or TPUs, suitable for data science analysis, such as Google Colab, Paperspace Gradient or Kaggle. In addition, the use of other pure cloud computing services[5] (not limited to notebooks, but also other options such as Azure ML jobs) was thought of but discarded due to their payment plan. Google Colab, defined below, in its pro tier was chosen as the working environment to carry out the experiments:

> *Google Colaboratory is a research project for prototyping machine learning models on powerful hardware options such as GPUs and TPUs. It provides a serverless Jupyter Notebook environment for interactive development [166].*

In this work, we used GPUs instead of TPUs. Although depending on demand, Colab can allocate different GPUs, most of the time Nvidia Tesla T4 (16GB, CUDA version 12.0) was assigned. Other GPUs such as Nvidia P100, V100, K80; are available and their availability also varies depending on the payment plan. 14 GB of RAM memory were used (Not a high-RAM runtime). Under this scenario (Nvidia Tesla T4 and 14 GB RAM), Google Colab estimates 1.96 compute unit cost per hour. A Google Colab Pro plan contains 100 compute units and has a cost of 11.19€, while a Google Colab Pro + plan contains 500 compute units and has a cost of 51.12€. Both plans were hired in this work depending on the demand of the tasks.

[4] on December, 11th

[5] https://cloud-gpus.com/

4.4.2.2 External validation tools

To fulfil the legal requirements and the General Data Protection Regulation (GDPR), the external validation with clinical notes from the HCSC is made locally. For that purpose, once the models are trained with Google Colab, they are downloaded, and the inference is performed locally. Due to the lack of computational resources, fine-tuning is not intended and only inference is performed.

The GPU used for inference is Apple M1 Pro, with 16 cores. Special caution should be given as Apple's Metal Performance Shaders is used as a backend for PyTorch and TensorFlow rather than CUDA. A list of available backends can be found in PyTorch webpage.

Chapter 5

System architecture and development phases

The workflow followed in this Master's thesis can be seen in Figure 5.1. In this chapter, all but the evaluation phase will be discussed. The code written to conduct the experiments is accessible through GitHub.

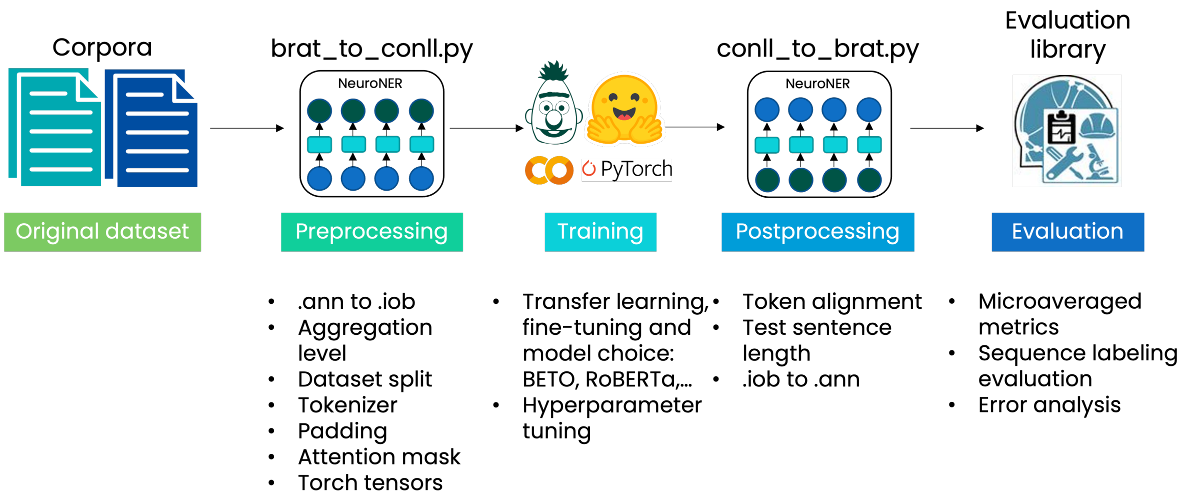

Figure 5.1: 5-phases workflow followed in this Master's thesis

5.1 Pre-processing

BERT based models are characterised for their low pre-processing burden and a performance decrease when applying typical NLP pre-processing steps such as stemming or stopwords removal. Several reasons contribute to this phenomenon: (i) BERT uses all of the information in a sentence, including punctuation and stopwords, (ii) BERT uses WordPiece tokenization to shrink its vocab size, (iii) de-capitalization is taken into account using BERT cased or uncased variants, (iv) the attention mechanism minimises the noise introduced by high-frequency words without the need to remove them. According to the Tensorflow documentation, pre-processing includes the following tasks: *"tokenizing text into subword units, combining sentences, trimming content to a fixed size and extracting labels for the masked language modelling task"*. As seen, most of the pre-processing tasks are oriented to transform the annotated data into the expected BERT input format.

1. .ann to .iob: *brat_to_conll.py* script from NeuroNER is used to transform the annotations in standoff BRAT format to BIO [167]. This script requires four parameters: i) path with '.txt' and '.ann' files, ii) path to output a single BIO file with all the annotations. This

file originally contains five columns and as many rows as tokens. The first column, *words* indicates the token, the second, *fileId*, indicates the file from which the token comes, the third and four, *start* and *end* columns indicate the start-offset and end-offset of the tokens. Finally, an additional column, *sentenceID*, indicates the phrase number, within a note, in which the token is located. This column was manually created. iii) the tokenizer, with the choice between spaCy or stanford and, iv) the language of the tokenizer. As suggested by the official spaCy documentation, es_core_news_sm was employed.

The following warning was obtained after running the script: *the text of the token contains space character, replaced with hyphen.* This was due to the space in "EE. UU." token. Therefore, this entity was transformed to "EE.-UU." with a hyphen. (See clinical note *cc_covid99.txt*). A pre-processing step had to be done for MOD corpus, as the annotations for both tasks (i.e., MEDDOPROF-NER and MEDDOPROF-CLASS) were contained in the same .ann file. A script for splitting the annotations in each .ann file into two .ann files (one for each task), was developed, *ProcesadoMOD.ipynb*.

2. Aggregation level: Different approaches to handle the length of the input text, and the maximum length of the BERT models are discussed in Section 2.4.2. In this work two alternatives are explored, based on the aggregation level, at the clinical note or at the sentence level:

   - Aggregation at the clinical note level: the whole clinical note is used as input to the model. As most of the clinical notes were longer than the maximum length allowed by BERT, they were truncated to a fixed length, with subsequent loss of information. The main benefit from this approach is a longer context.
   - Aggregation at the sentence level: The clinical notes were split into independent sentences and the models were trained with all the information contained in the clinical note. The length of the input sentences was defined after analysing the mean, median, and quartiles of the number of tokens per sentence.

3. Dataset split: The original training dataset is split into two subsets, training and validation, according to a fraction value.

4. Tokenizer, BERT special tokens creation, padding, masking, and torch tensors: The input data is tokenized according to the tokenizer implemented by the chosen model. After tokenization, the subtokens receive the same BIO tag that the original unsplit token. Besides, as the input text can be of varying lengths, padding is done to homogenize the length of all of them. Next, attention masks are created to ignore padding labels. Finally, the data are converted to torch tensors.

5.2 Training

First of all, the trained models belong to a supervised learning problem, more concretely, to a multiclassification task. In this scenario, the algorithm is trained with labelled data and the implemented solution tries to assign labels to data not previously seen. Different design decisions and hyperparameters are considered during the training phase, see Table 5.1. A distinction between design parameters (i.e., parameters that are considered specifically in this work, that can change the size of the training set, the task to be performed or the input data) and neural network parameters (e.g., regularisation, learning rate) is made. Depending on the combination of the design parameters, different models are trained and evaluated, following the hierarchy shown in Figure 5.2:

1. Task: NER and class MEDDOPROF subtasks are addressed independently in this work. However, other approaches described in Section 3.4.1 addressed them as a single joint task.

2. Training data corpus: as EdIE team did with ProfNER, see Chapter 3.4, we tried to expand the original MEDDOPROF training set with MOD corpus. Separate models are trained considering only MEDDOPROF data or in combination with MOD.

3. Aggregation level: to study the impact of attention and truncation, models are trained considering the clinical note as a whole and truncating the excess text, or using independent sentences. In both cases the evaluation is performed at the sentence level, this is, the test note is split, the inference is made and then all the sentences are merged.

4. Model: to study the impact of using general-domain pre-trained models and how their performance competes with specific-domain pre-trained models, BETO cased/uncased, ALBETO, DistilBETO and RoBERTa base biomedical clinical es models [88] are trained.

On the other hand, the hyperparameter values choice is based on the methodology and the results of the participant MEDDOPROF teams, already reviewed in Section 3.4.1. Moreover, a paper that discusses general training tips for the transformer model can be found in [168]. The hyperparameters were initialised as follows:

1. Fraction of training and validation data: the original training data is split into two sets containing the 80%, training, and the 20%, validation/development, of the data. This split criterion is the most commonly used, however, other splits could be considered depending on the amount of data (e.g., 90%-10%). Strategic datasplits as suggested by the NLNDE team, could also be used to boost the model's performance.

2. Optimizer: AdamW optimizer [169] was chosen as the default option. AdamW (i.e., Adam weight decay) is a variant of the Adam optimizer that implements a weight decay regularisation technique to prevent overfitting during training improving the generalization ability of the model. In models with a large number of parameters that require significant computational resources to train, reducing the likelihood of overfitting is crucial, so regularisation techniques are recommended. Hence, the use of AdamW is a commonly chosen option.

3. Maximum sentence length: As described in Chapter 2, traditional BERT models (including BETO) have a maximum length of 512 subword tokens. In addition, longer input sentences, have longer computing time due to increasing complexity (i.e., quadratic computational complexity). As only a few sentences had more than 512 subtokens, we set the maximum length to 510 to consider all the potential information contained in the text. When using the sentence as the level of aggregation rather than the whole clinical note, a shorter sequence length could be chosen.

4. Batch size: training large models can pose challenges, even on GPUs, due to their immense size, often leading to memory limitations and extended training times. Small batch sizes can fill up GPU memory, while a larger batch size can yield faster model convergence. Initially, we fix the batch size to 4.

5. Epochs: BERT authors recommend fine-tuning for 4 epochs. However, we set the value to 10 and plotted the train and validation learning curves.

6. Learning rate, gradient clipping, and epsilon: a common starting point is to use a learning rate value in the range of 2e-5 to 5e-5, gradient clipping value of 1 and epsilon value of 1e-8.

Additional models are trained by varying the learning rate and the batch size.

Finally, Cross-Validation (CV) was not used for hyperparameter fine-tuning for the following reasons: high number of training examples, transfer-learning good performance and computational costs (i.e., a single execution takes hours).

5.3 Post-processing

Once the models are trained, inference over the test set is intended. To achieve this goal, it is necessary to apply the same tokenizer used during training to the set of test notes. After that, the trained models are applied to the test notes and a classification label is retrieved per token. This label follows the BIO schema. However, three major concerns remain.

Table 5.1: Parameters considered in this work

Parameter	Description
	Design parameters
Task	MEDDOPROF subtask: identification of occupations or identification of the person to whom the occupation belongs
Data	Whether the model is trained using MEDDOPROF or MEDDOPROF + MOD corpus
Aggregation level	Use full clinical notes or single sentences as input
Model	Pre-trained transformer model used for fine-tuning. Cased and uncased variants are also considered within this parameter
	Other parameters
Training and validation split	Fraction of training instances to be retained in training and validation sets
	Neural network parameters (hyperparameters)
Optimizer	Component that updates the parameters of the model during training to minimise the loss function
Maximum length	Maximum length of the input data
Bath size	Number of training samples used in a single forward/backward pass of a neural network during training
Epochs	Number of times the complete set of training examples is presented to the model during training
Learning rate	Step size at which the model's parameters are updated during training
Gradient clipping	Regularization technique that limits the maximum size of the gradient by clipping its norm to a predefined threshold value
Epsilon	Small constant value that is added to the denominator of a numerical calculation to avoid division by zero

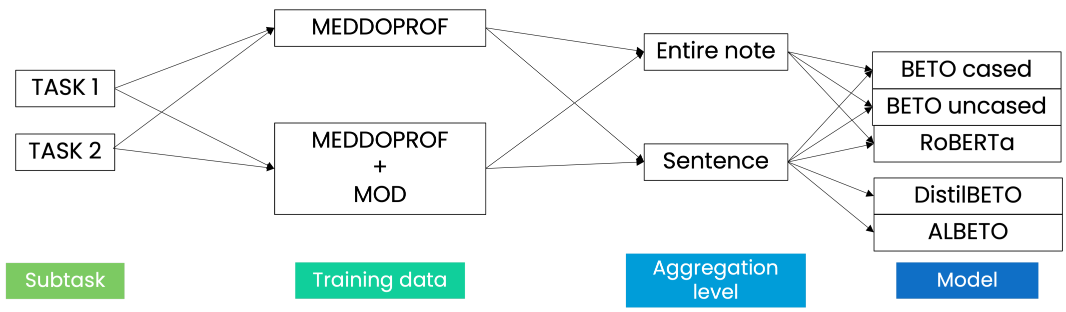

Figure 5.2: Training design paths

1. Token alignment: the first concern is explained in Section 2.2.1. Given that BERT uses WordPiece tokenization, some punctuation characters, originally belonging to previous tokens, form a token by themselves when applying tokenization as a preliminary step to token classification. This produces an alignment shift affecting the start- and end-offset of entities. This shift hampers the evaluation task, as the span of the recognised entities is not aligned with the entities of the gold standard. A list of characters found in the test set that can be attached to previous tokens or to the following tokens can be seen in Figure A.4. It is important to highlight that other tokenizers such as RoBERTa implement a method before splitting the word into tokens to handle spaces before words. This method consists of replacing spaces with $\dot{G}$ character to avoid digesting spaces. This could be helpful to preserve token alignment.

2. The second concern is related to the length of test sentences. If the length of the test sentences is greater than the maximum length with which the model has been trained, an error will raise: *"Token indices sequence length is longer than the specified maximum sequence length for this BERT model (length of test sentence > maximum length). Running this sequence through BERT will result in indexing errors"*

3. The third concern is related to the output format of the predictions, BIO. These predictions need to be parsed to .ann standoff format, because this is the one used in the gold standard.

The first issue could be addressed in several ways, such as training an own BERT implementation, or writing a post-processing script to re-align the NER tags. However, another approach was chosen in this work. Taking advantage of a parameter of transformers.BatchEncoding class, words_ids, a list of indices that indicates which tokens come from the same word is retrieved. This list is only generated when using the so-called fast tokenizers. Therefore, AutoTokenizer.from_pretrained is used with the parameter use_fast=True rather than BertTokenizer.from_pretrained. In addition, is_split_into_words parameter is set to true when applying the tokenizer. For the tokeniser to work, the input text should be stored in a list of strings. Finally, the predictions are given considering the positions of the list. The start-offset and end-offset can then be computed by counting characters and taking into account that some characters (e.g., such as commas) belong to the previous token, and some others (e.g., parentheses) to the following token.

The second concern could be addressed using different approaches: i) training a model able to handle longer sequences, ii) using a model not limited to sequence length such as XLNet, 3) splitting the text on which inference is to be drawn into smaller fragments that fit with the model length, make predictions and reassemble the fragments into the original text. However, some context may be lost depending on how the text is cut. Finally, the last approach and the one considered in this work is to truncate the sentence on which inference is to be made to the maximum length, and then assume that the rest of the tokens belong to the majority class "O" (i.e., no class prediction). This approach was chosen as only two sentences out of 344 in the test set contained over 510 tokens (i.e., 890 and 792).

The last issue is addressed using the function conll_to_brat.py from NeuroNER. This function was slightly modified (i.e., line 142 was commented) to adapt it to our data. Basically, the function receives four parameters: i) path to conll file to convert to BRAT annotations, ii) path to output conll file with filename and offsets that are compatible with BRAT annotations, iii) folder that contains the original .txt (and .ann) files that are formatted according to BRAT and iv) folder to output the text and BRAT annotations. With the modification proposed the first two parameters referred to the same file.

Moreover, special consideration should be given to uncased models since tokens are converted to lowercase. When comparing the predictions to the gold standard, there might be a mismatch since the lower-case version of a token (i.e., prediction) can be compared to the upper-case version of the same token (i.e., gold standard). Finally, [PAD] predicted tokens were converted to "O" tokens.

Chapter 6

System evaluation

6.1 Evaluation metrics

The evaluation metrics used are the standard ones employed in other NER tasks, precision, recall and F1-score, which are defined as follows:

$$Precision(P) = \frac{TP}{(TP + FP)}$$

$$Recall(R) = \frac{TP}{(TP + FN)}$$

$$F_\beta = (1 + \beta^2) * \frac{(P * R)}{\beta^2(P + R)}$$

When $\beta = 1$ (i.e., *Precision and Recall* receive the same attention):

$$F_1 = 2 * \frac{(P * R)}{(P + R)}$$

Where TP stands for *True Positive*, FP stands for *False Positive* and FN for *False Negative*. These terms came from the confusion matrix, shown below:

Prediction outcome

		p	**n**	**total**
Actual value	**p′**	True Positive	False Negative	P′
	n′	False Positive	True Negative	N′
	total	P	N	

Precision can also be seen as the relation between the number of correctly predicted tokens with respect to the number of predicted tokens. On its behalf, *recall* can be seen as the relationship between the number of tokens correctly predicted and the number of tokens in the dataset. Finally, *F1-score* is the harmonic mean of precision and recall. When evaluating the performance of a multiclassifier, the balancing between classes should be considered. In NER tasks following an BIO scheme, the *outside* entity will be the majority class.

All the metrics calculated are micro-averaged (i.e., metrics are computed ignoring entity types). More details of this last term can be found in [170]. This metric was chosen as the MEDDOPROF task organisers proposed it as the official evaluation metric for both tasks. In short, the micro-average aggregates the contributions of all classes to compute the average metric. This metric gives equal weight to each document of each class in a multiclass classification system. Therefore, the largest class would benefit, and more instances will be correctly classified. Conversely, the macro-average is encouraged to recognise every class correctly. When computing the macro-average, the metrics are computed per entity type and then averaged. To sum up, according to [47]:

> *In macroaveraging, we compute the performance microaveraging for each class and then average over classes. In microaveraging, we collect the decisions for all classes into a single confusion matrix, and then compute precision and recall from that table [...] microaverage is dominated by the more frequent class, since the counts are pooled. The macroaverage better reflects the statistics of the smaller classes, and so is more appropriate when the performance of all the classes is equally important.*

The shared task organisers provided a script, written in Python 3.8, to compute MEDDOPROF evaluation metrics. This script can be downloaded through GitHub. It is assumed that there are no completely overlapping annotations and that the prediction files are in BRAT standoff format. The steps taken to evaluate the predictions are:

1. Download the MEDDOPROF evaluation library
2. Store the predictions, both .ann and .txt files in a folder.
3. Store the gold standard test set (i.e., 344 notes) in a folder
4. Launch the script and specify the task (i.e., NER or CLASS). Pandas 1.2.4 is required to run successfully the script.

python main.py -g ../gold-standard-directory/ -p ../prediction-directory/ -s ner

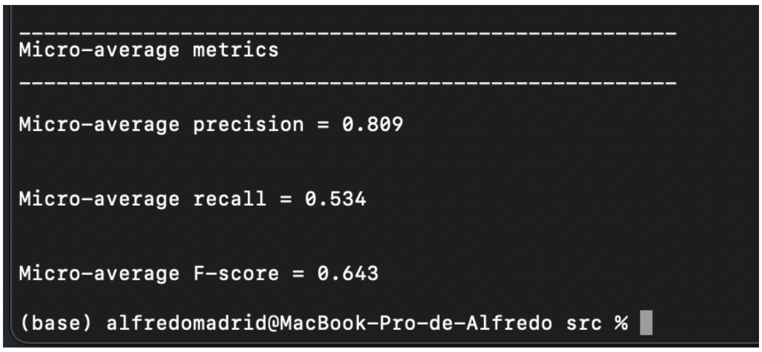

Figure 6.1: Evaluation script output

Other resources facilitated by task organisers can be seen in the Supplementary Table A.1

6.2 Results

Forty models were finally trained, eighteen for the first task (i.e., NER), occupation detection, and the rest for the second task (i.e., CLASS), to whom the occupation belongs. Table 6.1 shows the models' features and the evaluation metrics of all of them, using the script provided by the task organisers. The experiments were designed "on the fly", this is, on the basis of the results obtained so far. The following results are immediately noticeable from that table:

(i) Training at the sentence level is better than with the whole clinical note at once

(ii) Adding the MOD corpus to the training data worsens the results

(iii) Uncased BERT models have a lower performance than cased versions

(iv) The performance of task 2 is better than task 1, this is, is easier to build a model capable of recognising to whom the occupation belongs than a model for detecting occupations

Moreover, the best-performing result is obtained with the model #13, which uses RoBERTa pre-trained on biomedical data [88]. This is not surprising, as this model was trained with specific biomedical and clinical data.

Table 6.1: Results table

N	Design Decisions			Hyperparameters			TASK1-NER			TASK2-CLASS		
	Corpus	**Aggregation level**	**Model**	**Lr**	**Batch size**	**Epochs**	**P**	**R**	**F1**	**P**	**R**	**F1**
1	MEDDO	Sentence	BERT(c)	2E-05	4	10	0.809	0.534	0.643	0.759	0.709	0.733
2			BERT(u)	2E-05	4	10	0.788	0.527	0.631	0.748	0.686	0.716
3			RoBERTa	2E-05	4	10	0.836	0.547	0.661	0.741	0.724	0.732
4		Whole note	BERT(c)	2E-05	4	10	0.743	0.523	0.614	0.583	0.653	0.616
5			BERT(u)	2E-05	4	10	0.688	0.492	0.573	0.604	0.591	0.597
6			RoBERTa	2E-05	4	10	0.779	0.474	0.589	0.69	0.535	0.602
7	MEDDO + MOD	Sentence	BERT(c)	2E-05	4	10	0.831	0.532	0.649	0.75	0.672	0.709
8			BERT(u)	2E-05	4	10	0.829	0.506	0.628	0.709	0.654	0.68
9			RoBERTa	2E-05	4	10	0.824	0.518	0.636	0.737	0.699	0.717
10		Whole note	BERT(c)	2E-05	4	10	0.669	0.524	0.588	0.659	0.582	0.618
11			BERT(u)	2E-05	4	10	0.727	0.483	0.581	0.669	0.585	0.624
12			RoBERTa	2E-05	4	10	0.769	0.455	0.572	0.676	0.582	0.625
13	MEDDO	Sentence	RoBERTa	2E-05	8	10	**0.833**	**0.553**	**0.664**	**0.749**	**0.735**	**0.742**
14			RoBERTa	5E-05	8	10	0.759	0.509	0.61	0.7	0.681	0.691
15			RoBERTa	2E-05	8	4	0.811	0.55	0.655	0.709	0.712	0.71
16			RoBERTa	2E-05	4	4	0.81	0.543	0.65	0.741	0.726	0.734
17			ALBERT	2E-05	4	10	0.789	0.526	0.631	0.731	0.698	0.714
18			DistilBERT	2E-05	4	10	0.781	0.52	0.625	0.709	0.685	0.697
19	MEDDO	Sentence	BERT(c)	2E-05	8	10	0.815	0.554	0.66	0.762	0.723	**0.742**
20	MEDDO + MOD	Sentence	RoBERTa	2E-05	8	10	0.825	0.532	0.647	0.722	0.733	0.727

MEDDO: MEDDOPROF, c: cased, u: uncased, Lr: Learning rate, P: Precision, R: Recall. All models were trained with Tesla T4 GPU, eps = 1E-08, Max length = 510, Max grad norm = 1, Optimizer = AdamW. All models are the Spanish adaptation of the original ones. The RoBERTa model is pre-trained on a biomedical-clinical corpus.

The prevailing metric in all models, when studying TASK1-NER, is precision over recall. That is, the models exhibit proficiency in classifying entities, albeit leaving some entities undiscovered. In TASK2-CLASS, the relation between these two metrics is more balanced.

The average training time with a Tesla T4 GPU can be seen in Table 6.2. The results are as expected:

(i) When adding the MOD corpus, the training time is extended by approximately 2 hours.

(ii) When training at the whole note level, rather than at the sentence level, there are fewer data due to the padding to the maximum length, therefore the models train considerably faster.

(iii) The training time of distilled models is much lower than non-distilled models without unduly compromising performance.

The best-performing model was trained for 12h 13min 51s. Learning curves for all the models were also plotted, Figure 6.2 shows the model #13 learning curves. In that figure, it can be appreciated that the training and validation loss stay close for the first 2 epochs and then start to slowly diverge. The best-performing model obtained is ranked 10/17 and 4/13, when compared to the results shown by the MEDDOPROF participant teams, for the first and second tasks, respectively, Figure 6.3.

Table 6.2: Average training time with a Nvidia Tesla T4 GPU

Model	Training time
1, 2, 3, 13, 14	13h 20min
4, 5, 6	25 min
7, 8, 9	15h 30 min
10, 11, 12	30 min
15	5h
16	5h 20 min
17, 19	11h 35 min
18	6h 50 min
20	14h 45 min

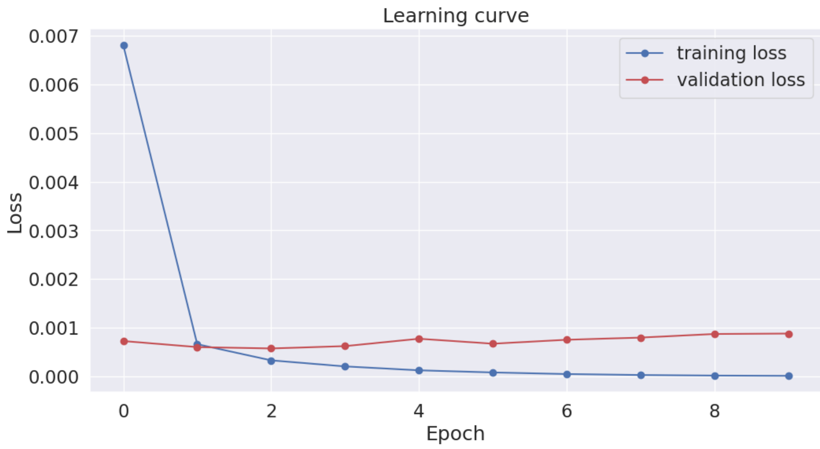

Figure 6.2: Training and validation loss

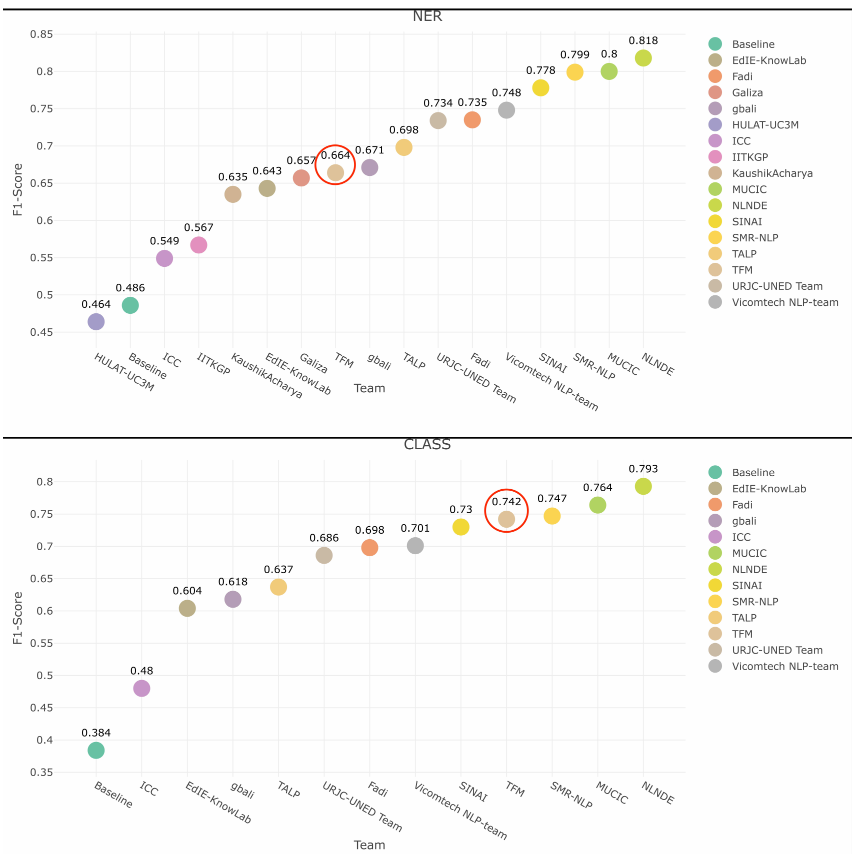

Figure 6.3: Performance of the best-performing solution compared to other solutions presented in the MEDDOPROF shared task

6.3 Error analysis

To begin with, in 140 and 150 out of 344 test set notes from TASK1 and TASK2, there is at least one classification error. The confusion matrix for both tasks can be seen in Table 6.3 and Table 6.4, respectively.

Table 6.3: TASK1-NER confusion matrix

		Actual							
		B-ACT	B-PRO	B-SIT	I-ACT	I-PRO	I-SIT	O	support
Predicted	B-ACT	13	4	3	0	0	0	8	**28**
	B-PRO	6	630	5	1	9	0	45	**696**
	B-SIT	0	8	254	0	0	15	80	**357**
	I-ACT	0	0	0	23	1	1	35	**60**
	I-PRO	0	4	0	3	998	5	134	**1144**
	I-SIT	0	0	7	0	6	305	175	**493**
	O	11	20	55	29	88	103	212447	**212753**
	total predicted	**30**	**666**	**324**	**56**	**1102**	**429**	**212924**	**215531**

ACT: Actividad (activity), PRO: Profesión (profession), SIT: Situación laboral (working status)

The confusion matrix in Table 6.3 shows that most of the errors are due to the system omitting entities rather than to misclassification. 66 *profession*, 103 *working status* and 15 *activities* entities were not recognised by our system.

In addition, more than half of the *activity* labels are misclassified and are assigned to *profession* and *O* entities. This is caused by the imbalanced and the low number of *activity* samples. The system proposed for TASK1 underestimates the number of tokens of all the entities except for *B-ACT* entity. Around 90% of *profession* entities are correctly classified. Nevertheless, the number of correctly identified entities for *working status* decreases to 60-70%. This might be explained by the fact that the *profession* is the majority entity

The summary that could be given of this confusion matrix is that the errors are mostly due to the non-identification of entities rather than errors between different entity types.

Table 6.4: TASK2-CLASS confusion matrix

		Actual								
		B-FAM	B-OTROS	B-PACI	B-SAN	I-FAM	I-OTROS	I-PAC	I-SAN	O	support
Predicted	B-FAM	34	0	8	0	1	0	0	0	9	**52**
	B-OTROS	3	100	5	12	0	0	0	0	26	**146**
	B-PAC	9	7	432	4	0	0	29	0	109	**590**
	B-SAN	0	2	0	284	0	0	0	3	4	**293**
	I-FAM	1	0	1	0	51	0	19	0	15	**87**
	I-OTROS	0	0	0	0	0	59	4	0	17	**80**
	I-PAC	0	0	13	0	17	11	890	13	287	**1231**
	I-SAN	0	0	0	0	0	0	0	283	16	**299**
	O	4	16	67	2	15	20	199	17	212413	**212753**
	total predicted	**51**	**125**	**526**	**302**	**84**	**90**	**1141**	**316**	**212896**	**215531**

FAM: Familiar (family member), PAC: Paciente (patient), SAN: Sanitario (health professional)

This confusion matrix, Table 6.4, shows a similar behaviour to the previous one. To begin with, all the entities are underestimated. Therefore, the misclassification error between entities is not as common as system-omitting errors. The impact of the imbalance is also present here. The least predominant classes are *family* and *others*. The best-predicted class is *health professional* (i.e., SAN). This may be because the entities are used in similar contexts within the medical records and the entities are similar: "Dr.", "MAP", "AUX", "enfermera" and so on. Surprisingly, the token that accounts for the majority of omissions (i.e., is misclassified in O tokens) is "I-PAC". This is, the system tends to fail in the detection of the boundaries of the *patient* entity.

For more in-depth analysis, the seqeval library was used. With this library metrics at the entity level are provided. See Table 6.5 and 6.6. Note that the microaveraged F1-score differs from the one obtained with the official MEDDOPROF evaluation task script. It is worth noting that the minority classes have the lowest F1 value as discussed earlier.

Table 6.5: TASK1-NER results according to seqeval library

	P	**R**	**F1**	**support**
ACT	0.27	0.36	0.31	28
PROF	0.87	0.85	0.86	696
SIT	0.68	0.66	0.67	357
micro avg	0.79	0.78	0.78	
macro avg	0.61	0.62	0.61	1081
weighted avg	0.79	0.78	0.78	

ACT: Actividad (activity), PRO: Profesión (profession), SIT: Situación laboral (working status)

Table 6.6: TASK2-CLASS results according to seqeval library

	P	**R**	**F1**	**support**
FAM	0.53	0.60	0.56	52
OTROS	0.75	0.66	0.70	146
PAC	0.70	0.68	0.69	590
SAN	0.90	0.94	0.92	293
micro avg	0.75	0.74	0.75	
macro avg	0.72	0.72	0.72	1081
weighted avg	0.75	0.74	0.75	

FAM: Familiar (family member), PAC: Paciente (patient), SAN: Sanitario (health professional)

The scikit-learn library was also used to obtain metrics at the BIO level for both tasks, Table 6.7 and Table 6.8. However, the use of this library in NER tasks is discouraged because it considers entities as independent subjects. The insights that can be drawn from the previous two tables reinforce the idea of imbalance-low F1 score relation.

A third additional evaluation library, nervaluate, was used and the results can be seen in Table A.7 and Table A.8. This library provides five metrics to consider different categories of errors and four different ways to measure errors, with varying degrees of strictness. For a more detailed description of such classification, the reader is encouraged to read Section A.8.

Table 6.7: TASK1-NER results according to scikit-learn library

	P	**R**	**F1**	**support**
B-ACT	0.43	0.46	0.45	28
B-PROF	0.95	0.91	0.93	696
B-SIT	0.78	0.71	0.75	357
I-ACT	0.41	0.38	0.40	60
I-PROF	0.91	0.87	0.89	1144
I-SIT	0.71	0.62	0.66	493
O	1.00	1.00	1.00	212753

ACT: Actividad (activity), PRO: Profesión (profession), SIT: Situación laboral (working status)

Table 6.8: TASK2-CLASS results according to scikit-learn library

	P	R	F1	support
B-FAM	0.67	0.65	0.66	52
B-OTROS	0.80	0.68	0.74	146
B-PAC	0.82	0.73	0.77	590
B-SAN	0.94	0.97	0.95	293
I-FAM	0.61	0.59	0.60	87
I-OTROS	0.66	0.74	0.69	80
I-PAC	0.78	0.72	0.75	1231
I-SAN	0.90	0.95	0.92	299
O	1.00	1.00	1.00	212753

FAM: Familiar (family member), PAC: Paciente (patient), SAN: Sanitario (health professional)

6.3.1 Error examples

The errors made by the models are classified into the following categories:

- Success:
  1. Exact match: the complete set of tokens predicted by the model/system corresponds to the set of tokens annotated by the experts in the test set, this is, a hit.
- Mistake:
  2. Partial match: the model predicted the entity but its span or boundaries are not correctly identified. This is, some entity tokens have been predicted by the model but not the whole entity.
  3. False positive (type I error): the model incorrectly identifies a word or phrase as an entity when it is not.
  4. False negative (type II error): the model fails to identify an entity that is present in the text.
  5. Misclassification: the model assigns the wrong entity type to a word or phrase.
  6. Misclassification and partial match (2 and 5 cases at the same time): the model assigns the wrong entity type to a word or phrase and the span or boundaries are not correctly identified.

Some examples of errors made by the model, for both tasks NER and CLASS, are shown in Table 6.9 and Table 6.10 respectively.

Table 6.9: Examples of error types produced by the best-performing model in TASK1-NER

N	Error type	Token	Golden standard	Prediction	Clinical note
1	Partial match	Trabaja	O	B-PROF	S1130 52742017000100001-1
		en	O	I-PROF	
		el	O	I-PROF	
		ámbito	O	I-PROF	
		militar	B-PROF	I-PROF	
2	Partial match	ex	B-PROF	B-PROF	caso clinico medicina interna1242
		trabajadora	I-PROF	B-PROF	
		de	I-PROF	I-PROF	
		fábrica	I-PROF	I-PROF	
		textil	I-PROF	I-PROF	

3	Partial match	Aparición	O	O	caso clinico medicina interna888
		hematoma	O	O	
		mientras	B-SIT	O	
		trabajaba	I-SIT	B-SIT	
4	Partial match	ha	B-SIT	I-SIT	caso clinico psiquiatria14
		repetido	I-SIT	I-SIT	
		2	I-SIT	I-SIT	
		cursos	I-SIT	I-SIT	
		de	I-SIT	I-SIT	
		ESO	I-SIT	I-SIT	
5	False positive	Calle	O	O	S0034 98872012001100010-1
		del	O	O	
		Alcalde	O	B-PROF	
		Francisco	O	O	
		Santero	O	O	
6	False positive	No	O	O	caso clinico atencion primaria104
		pudo	O	O	
		despedirse	O	B-SIT	
		de	O	I-SIT	
		ella	O	I-SIT	
		por	O	O	
		problemas	O	O	
		económicos	O	O	
7	False positive	había	O	O	caso clinico dermatologia456
		comenzado	O	I-SIT	
		a	O	O	
		trabajar	O	O	
8	False positive	mantenga	O	O	caso clinico psiquiatria14
		durante	O	O	
		meses	O	O	
		un	O	I-SIT	
		trabajo	O	I-SIT	
9	False positive	Estudios	O	O	caso clinico psiquiatria220
		hasta	O	O	
		3o	O	O	
		de	O	I-SIT	
		BUP	O	I-SIT	
10	False negative	trabajó	B-PROF	O	S1132 62552015000100005-1
		en	I-PROF	O	
		otro	I-PROF	O	
		centro	I-PROF	O	
		sanitario	I-PROF	O	
11	False negative	Cuida	O	O	caso clinico atencion primaria3
		de	O	O	
		sus	O	O	
		padres	O	O	
		,	O	O	
		dependientes	B-SIT	O	
		,	O	O	
12	False negative	También	O	O	caso clinico psiquiatria14
		simultanea	B-SIT	O	
		trabajos	I-SIT	O	

13	False negative	dedica	B-ACT	O	caso clinico psiquiatria14
		muchas	I-ACT	O	
		horas	I-ACT	O	
		a	I-ACT	O	
		ir	I-ACT	O	
		al	I-ACT	O	
		gimnasio	I-ACT	O	
14	False negative	levantarse	O	O	caso clinico psiquiatria163
		por	O	O	
		las	O	O	
		mañanas	O	O	
		para	O	O	
		ir	B-SIT	O	
		al	I-SIT	O	
		colegio	I-SIT	O	
15	Misclassification	32	O	O	S0034 98872006000200011-1
		años	O	O	
		,	O	O	
		deportista	B-PROF	B-ACT	
		,	O	O	
16	Misclassification	Trabajo	B-SIT	B-PROF	caso clinico psiquiatria220
		de	I-SIT	I-PROF	
		baja	I-SIT	I-PROF	
		cualificación	I-SIT	I-PROF	
17	Misclassification	calidad	O	O	caso clinico psiquiatria382
		de	O	O	
		vida	O	O	
		de	O	O	
		los	O	O	
		familiares	O	O	
		cuidadores	B-PROF	B-SIT	
18	Misclassification	debe	O	O	caso clinico psiquiatria474
		abandonar	B-ACT	B-SIT	
		la	I-ACT	I-SIT	
		práctica	I-ACT	I-ACT	
		del	I-ACT	I-ACT	
		fútbol	I-ACT	I-ACT	
19	Misclassification	Interina	B-PROF	B-SIT	casos clinicos profesiones127
		en	I-PROF	I-SIT	
		una	I-PROF	I-SIT	
		casa	I-PROF	I-SIT	
20	Misclassification + partial match	Un	O	O	casos clinicos infecciosas53
		compañero	B-ACT	B-SIT	
		de	I-ACT	O	
		viaje	I-ACT	O	
21	Misclassification + partial match	Tocó	B-PROF	B-ACT	casos clinicos profesiones24
		la	I-PROF	I-ACT	
		guitarra	I-PROF	I-ACT	
		en	I-PROF	0	
		varios	I-PROF	0	
		conciertos	I-PROF	0	

Some interesting findings from the previous table are as follows:

- The model has difficulties in correctly delimiting the entities as shown in example #1. According to the MEDDOPROF annotation guidelines, *G12 Relevancia* rule this should be refined.
- The model fails when dealing with *ex* affix, as shown in example #2, and as described in *P15 Prefijos* rule.
- According to the MEDDOPROF guidelines, *"sufre mientras trabajaba en su huerta"*, *mientras* should not be annotated. However, in example #3 it is annotated in the gold standard. This seems to be an inconsistency between the annotation guidelines and the gold standard.
- Surprisingle, the model fails, on rare occasions, when establishing the BIO tag order. In example #4 the model correctly identified the entity, however, the first token given is "I-" rather than "B-".
- From example #5, it would be desirable for the model to see more training examples of streets containing professions. Similarly, more cases where the word "despedirse" is used as a form to say someone when you or they are leaving, should be provided to the model, as in example #6.
- In example #15, it is unclear from the original clinical note whether the patient is a proffesional athlete or not. Depending on this, the model would have failed or not.
- Example #18 is quite interesting, since the model fails combining two entity types, *working status* and *activity*.
- The last two examples, #20 and #21, are uncommon. The model does not recognise the correct entity and does not settle the correct boundaries, so two errors are made per entity.

A similar analysis is conducted with the data presented in Table 6.10.

Table 6.10: Examples of error types produced by the best-performing model in TASK2-CLASS

N	Error type	Token	Golden standard	Prediction	Clinical note
1	Partial match	la	O	O	32605766 ES
		enfermera	B-SAN	B-SAN	
		del	O	I-SAN	
		paciente	O	I-SAN	
		señaló	O	O	
2	Partial match	de	O	O	casos clinicos profesiones178
		celador-conductor	B-PAC	B-PAC	
		con	I-PAC	O	
		grado	I-PAC	O	
		de	I-PAC	O	
		discapacidad	I-PAC	O	
		reconocida	I-PAC	O	
3	Partial match	trabajadora	B-PAC	B-PAC	S1132 62552015000100005-1
		del	I-PAC	I-PAC	
		servicio	I-PAC	I-PAC	
		de	I-PAC	I-PAC	
		radiodiagnóstico	I-PAC	I-PAC	
		del	I-PAC	O	
		hospital	I-PAC	O	
4	Partial match	Trabaja	O	B-PAC	S1130 52742017000100001-1
		en	O	I-PAC	
		el	O	I-PAC	
		ámbito	O	I-PAC	

		militar	B-PAC	I-PAC	
5	Partial match	Actualmente	O	O	caso clinico atencion primaria146
		en	B-PAC	B-PAC	
		estado	I-PAC	I-PAC	
		de	I-PAC	O	
		incapacidad	I-PAC	B-PAC	
		temporal	I-PAC	I-PAC	
6	Partial match	ex	B-PAC	B-PAC	caso clinico medicina interna1270
		trabajador	I-PAC	B-PAC	
		de	I-PAC	I-PAC	
		minas	I-PAC	I-PAC	
		de	I-PAC	I-PAC	
		pirita	I-PAC	I-PAC	
7	False positive	TCAE	O	B-PAC	casos clinicos profesiones174*
		de	O	I-PAC	
		urgencias	O	I-PAC	
		hospitalarias	O	I-PAC	
8	False positive	convive	O	O	caso clinico psiquiatria417
		con	O	O	
		compañeras	O	B-OTROS	
		de	O	I-OTROS	
		piso	O	I-OTROS	
9	False positive	El	O	O	casos clinicos profesiones193
		paciente	O	O	
		retornó	O	B-OTROS	
		a	O	O	
		su	O	O	
		empleo	O	O	
		habitual	O	O	
10	False positive	Trabaja	O	O	casos clinicos profesiones199
		como	O	O	
		administrativa	B-PAC	B-PAC	
		y	O	O	
		también	O	O	
		realiza	O	B-PAC	
		reuniones	O	I-PAC	
11	False positive	Deseo	O	O	casos clinicos profesiones208
		de	O	O	
		reincorporación	O	B-PAC	
		laboral	O	O	
12	False positive	Ha	O	B-PAC	casos clinicos profesiones208
		conseguido	O	I-PAC	
		un	O	I-PAC	
		trabajo	O	I-PAC	
13	False negative	Cartera	B-PAC	O	caso clinico medicina interna1700
		de	O	O	
		profesión	O	O	
14	False negative	Su	O	O	S1130 52742017000100001-1
		familia	O	O	
		está	O	O	
		relacionada	O	O	
		con	O	O	

		el	O	O	
		ámbito	O	O	
		militar	B-FAM	O	
15	False negative	Cuida de sus padres , dependientes ,	O O O O O B-FAM O	O O O O O O O	caso clinico atencion primaria3
16	False negative	Realiza más deporte	B-PAC I-PAC I-PAC	O O O	caso clinico atencion primaria3
17	False negative	auxiliar de enfermeria (AXE)	B-PAC I-PAC I-PAC O B-PAC O	B-PAC I-PAC I-PAC O O O	casos clinicos profesiones149
18	Misclassification	Facultativo de área quirúrgica con	B-PAC I-PAC I-PAC I-PAC O	B-SAN I-SAN I-SAN I-SAN O	casos clinicos profesiones166
19	Misclassification	. Médico :	O B-SAN O	O I-SAN O	S1130 01082015000700010-1
20	Misclassification	que clasifica al trabajador en tres categorías	O O O B-OTROS O O O	O O O B-PAC O O O	S1132 62552015000100005-1 S1132 62552015000100005-1
21	Misclassification	. personal de servicios aeroportuarios	O B-PAC I-PAC I-PAC I-PAC	O B-OTROS I-OTROS I-OTROS I-OTROS	casos clinicos profesiones218
22	Misclassification + partial match	Enfermera del H.U.G.C.D.N.	B-PAC O O	B-SAN I-SAN I-SAN	casos clinicos profesiones172

Some interesting findings from the previous table are as follows:

- Examples #1-#5 show that the system fails equally by adding unnecessary tokens or by reducing the number of original tokens. No preference is shown.
- Example #2 is controversial. Two entities could have been considered instead of one. *celador-conductor*, which is a profession and *con grado de discapacidad reconocida* which is a working situation.
- Example #6 shows, once again, that the model fails to capture *ex* meaning.

- According to the gold standard, example #7 is a false positive. However, we strongly believe that this is an error from the test set. *TCAE* is an abbreviation of *técnicos en cuidados auxiliares de enfermería*, therefore the entity recognised by the model should be correct.
- Example #17 is remarkable. The model is able to recognise one profession, but not able to recognise its abbreviation, possibly because it has not seen similar cases.
- The model assumes that health professions belong to health workers. However, in #18 the health profession belongs to the patient. More training examples with this casuistry should be provided to the model.
- Example #22 is controversial. To begin with, it is similar to example #17. The model made the same assumption. However, according to the gold standard only *Enfermera* is part of an entity. According to the model, the whole entity is *Enfermera del H.U.G.C.D.N.*. Looking at example #3 and according to the gold standard *trabajadora del servicio de radiodiagnóstico del hospital* is a whole entity. This is due to rule N14 (no_sector), which states that no reference shall be made to whether the work activity is in the public or private sector.

The error analysis conducted so far could be used to refine the model by proposing training examples with the type of entities it tends to omit and misclassify. The code developed for the error analysis can be found at: `https://github.com/fredymad/TFM-UNED-DATOS/`

6.4 MOD corpus error analysis

To evaluate the performance of the MOD corpus, model #20 from Table 6.1 was developed. The same characteristics (i.e., hyperparameters and design decisions) as the best model obtained using only the MEDDOPROF corpus, model #13, were set. The purpose of this was to understand why performance was deteriorating when adding more data. The confusion matrices for both tasks can be seen in Tables 6.11 and 6.12.

When comparing the confusion matrices of the model generated with the MEDDOPROF corpus, #13 (Table 6.3), with the model generated with the MEDDOPROF and the MOD corpus, #20 (Table 6.11), for TASK1, it can be appreciated that the number of false positives for the *activity* entity decreases for model #20, while the number of true positives remains unchanged. The number of false positives for *profession* and *working status* entities decreases, but the number of true positives also decreases. In TASK-2, the number of true positives for the *family* and *health professional* entities increases (Table 6.12) when adding the MOD corpus, and when compared to model #13 (Table 6.4). The number of true positives for the *patient* entity remains unchanged, although the number of false positives increases. The opposite occurs with the entity type *others*, as the number of true positives decreases, but so does the number of false positives.

The aforementioned explanation is substantiated by the results derived from the seqeval library, see Table 6.13 and Table 6.14 respectively. The F1 for the *activity* entity increases in model #20, but decreases for the rest of entities. Similarly, increases for the *family* entity, remains unchanged for the *patient* entity and decreases for the rest. It is important to link the previous results with the proportion of entities added with the MOD corpus. As shown in Tables 4.3 and 4.7, the proportion of *activity* entities in the MOD corpus (i.e., 6.26%) doubles the proportion of *activity* entities in the MEDDOPROF corpus (i.e., 3.10%). A slight increase in the proportion of *family* entities can also be seen in the MOD corpus (i.e., 6.42% versus 5.5%).

For the first task, out of the 861 tags missed by model #13, model #20 correctly identifies 227 (26.84%). Conversely, out of the 941 tags missed by model #20, model #13 correctly identifies 307 (32.6%). For the second task, out of the 986 tags missed by model #13, model #20 correctly identifies 270 (27.38%). Contrarily, out of the 1,021 tags missed by model #20, model #13 correctly identifies 305 (29.87%).

Finally, examples of errors committed when adding the MOD corpus, but not made when using only MEDDOPROF can be seen in Tables 6.15 and 6.16.

Table 6.11: TASK1-NER confusion matrix when adding the MOD corpus

		Actual							
		B-ACT	B-PRO	B-SIT	I-ACT	I-PRO	I-SIT	O	support
Predicted	B-ACT	12	2	2	0	0	0	12	**28**
	B-PRO	6	600	13	0	10	4	63	**696**
	I-ACT	2	0	0	16	1	2	39	**60**
	I-PRO	0	6	0	3	994	4	137	**1144**
	I-SIT	0	1	5	0	9	304	174	**493**
	O	5	19	61	11	85	125	212447	**212753**
	total predicted	**25**	**637**	**298**	**30**	**1100**	**453**	**212988**	**215531**

ACT: Actividad (activity), PRO: Profesión (profession), SIT: Situación laboral (working status)

Table 6.12: TASK2-CLASS confusion matrix when adding the MOD corpus

		Actual									
		B-FAM	B-OTROS	B-PACI	B-SAN	I-FAM	I-OTROS	I-PAC	I-SAN	O	support
Predicted	B-FAM	36	1	5	0	1	0	0	0	9	**52**
	B-OTROS	5	97	5	11	0	0	0	0	28	**146**
	B-PAC	8	2	432	3	0	0	27	0	118	**590**
	B-SAN	0	1	0	282	0	0	0	2	8	**293**
	I-FAM	2	0	0	0	57	0	13	0	15	**87**
	I-OTROS	0	1	0	0	0	59	3	0	17	**80**
	I-PAC	0	0	19	0	14	6	932	8	252	**1231**
	I-SAN	0	0	0	0	0	0	0	292	7	**299**
	O	4	15	97	0	18	14	269	13	212323	**212753**
	total predicted	**55**	**117**	**558**	**296**	**90**	**79**	**1244**	**315**	**212777**	**215531**

FAM: Familiar (family member), PAC: Paciente (patient), SAN: Sanitario (health professional)

Table 6.13: TASK1-NER results according to seqeval library when adding the MOD corpus

	P	**R**	**F1**	**support**
ACT	0.40	0.43	0.41	28
PROF	0.85	0.82	0.84	696
SIT	0.58	0.56	0.57	357
micro avg	0.75	0.72	0.74	1081
macro avg	0.61	0.60	0.61	1081
weighted avg	0.75	0.72	0.74	1081

ACT: Actividad (activity), PRO: Profesión (profession), SIT: Situación laboral (working status)

Table 6.14: TASK2-CLASS results according to seqeval library when adding the MOD corpus

	P	**R**	**F1**	**support**
FAM	0.56	0.65	0.60	52
OTROS	0.76	0.62	0.68	146
PAC	0.64	0.67	0.66	590
SAN	0.90	0.94	0.92	293
micro avg	0.72	0.74	0.73	1081
macro avg	0.72	0.72	0.72	1081
weighted avg	0.72	0.74	0.73	1081

FAM: Familiar (family member), PAC: Paciente (patient), SAN: Sanitario (health professional)

Table 6.15: Example of errors made when adding the MOD corpus that did not appear with the MEDDOPROF corpus

N	Error type	Token	MEDDOPROF	MEDDOPROF + MOD	Clinical note
1	False positive	Trabaja de .	O O O	B-SIT I-SIT O	casos clinicos profesiones121
2	Partial match	La terapeuta referente del servicio de drogodependencia	O B-PROF I-PROF I-PROF I-PROF I-PROF I-PROF	O B-PROF O O O O O	caso clinico psiquiatria390
3	Misclassification + partial match	Médico activo	B-PROF O	B-PROF I-SIT	casos clinicos profesiones201
4	Partial match	Estudiante de FP	B-SIT O O	B-SIT I-SIT I-SIT	casos clinicos profesiones115
5	Misclassification + partial match	Chófer de ómnibus	B-PROF I-PROF I-PROF	B-ACT I-PROF I-ACT	caso clinico psiquiatria5
6	False negative	, percusionista ,	O B-PROF O	O O O	casos clinicos profesiones167

Table 6.16: Example of errors made when adding the MOD corpus that did not appear with the MEDDOPROF corpus. TASK2-CLASS

N	Error type	Token	MEDDOPROF	MEDDOPROF + MOD	Clinical note
1	False positive	En cuidados paliativos domiciliarios	O O O O	B-PAC I-PAC I-PAC I-PAC	casos clinicos profesiones221
2	Misclassification	La paciente está casada con un vendedor	O O O O O O B-FAM	O O O O O O B-OTROS	casos clinicos profesiones38
3	Partial match	retorno de incapacidad temporal	O O B-PAC I-PAC	O O O I-PAC	casos clinicos profesiones175
4	False negative	médico urgenciologo	B-SAN I-SAN	O O	casos clinicos profesiones117
5	False positive	solicita el alta laboral	O O O O	O O B-PAC I-PAC	caso clinico psiquiatria212

6.5 Performance of the models in the HCSC cohort

The best model, model #13, was trained with the entire MEDDOPROF corpus training set, comprising training and validation sets, and predictions were made on 2,000 clinical notes of the first visit of patients attending the HCSC during April 1st 2007 to November, 30th. Results are shown in Table 6.17.

Table 6.17: Results in the HCSC MediLog notes

Task	**P**	**R**	**F1**
NER	0.74	0.69	0.72
CLASS	0.70	0.74	0.72

At first glance, it can be seen that the results are better, for TASK-1, than the ones obtained in Table 6.1. This phenomenon may be attributable to two factors:

- Simpler notes with similar syntactic structure.
- High prevalence of entities readily identifiable by the model, such as 'Dr.', 'MAP', 'traumatólogo (orthopedic surgeon)', etc.

This is also noticeable in Table 6.18 and Table 6.19. As with the other confusion matrices, the system is prone to false negatives rather than to misclassification (this is, when the system makes an error, it is mainly because it omits to assign an entity to a word which is an entity).

Table 6.18: TASK1-NER confusion matrix in HCSC notes

		Actual							
		B-ACT	B-PRO	B-SIT	I-ACT	I-PRO	I-SIT	O	support
Predicted	B-ACT	12	2	0	5	0	0	9	**28**
	B-PRO	0	545	0	0	7	0	122	**674**
	B-SIT	0	2	33	0	0	2	17	**54**
	I-ACT	1	0	0	19	3	0	18	**41**
	I-PRO	0	1	0	0	299	3	37	**340**
	I-SIT	0	0	2	4	4	41	21	**72**
	O	35	96	28	55	84	41	200625	**200964**
	total predicted	**48**	**646**	**63**	**83**	**397**	**87**	**200849**	**202173**

Table 6.19: TASK2-CLASS confusion matrix in HCSC notes

		Actual									
		B-FAM	B-OTROS	B-PACI	B-SAN	I-FAM	I-OTROS	I-PAC	I-SAN	O	support
Predicted	B-FAM	1	0	0	5	0	0	0	0	0	**6**
	B-OTROS	0	2	0	0	0	0	0	0	0	**2**
	B-PAC	0	0	167	5	0	0	18	0	40	**230**
	B-SAN	0	1	4	423	0	0	0	0	90	**518**
	I-FAM	0	0	0	0	3	0	0	0	0	**3**
	I-OTROS	0	0	0	0	0	0	0	0	0	**0**
	I-PAC	0	0	4	0	0	0	318	7	57	**386**
	I-SAN	0	0	0	0	0	2	5	52	5	**64**
	O	1	6	99	74	0	2	172	21	200589	**200964**
	total predicted	**2**	**9**	**274**	**507**	**3**	**4**	**513**	**80**	**200781**	**202173**

When examining the F1-score at the entity level, *profession*, Table 6.20 and *health professional*, Table 6.21, are the best-recognised entities.

Table 6.20: TASK1-NER results according to seqeval library. HCSC notes

	P	**R**	**F1**	**support**
ACT	0.21	0.39	0.27	28
PROF	0.81	0.79	0.80	674
SIT	0.44	0.59	0.50	54
micro avg	0.74	0.76	0.75	756
macro avg	0.49	0.59	0.53	756
weighted avg	0.76	0.76	0.76	756

ACT: Actividad (activity), PRO: Profesión (profession), SIT: Situación laboral (working status)

Table 6.21: TASK2-NER results according to seqeval library. HCSC notes

	P	**R**	**F1**	**support**
FAM	0.50	0.17	0.25	6
OTROS	0.22	1.00	0.36	2
PAC	0.55	0.70	0.62	230
SAN	0.83	0.81	0.82	518
micro avg	0.72	0.77	0.75	756
macro avg	0.53	0.67	0.51	756
weighted avg	0.74	0.77	0.75	756

FAM: Familiar (family member), PAC: Paciente (patient), SAN: Sanitario (health professional)

In a clinical setting, it is of utmost importance to accurately identify entities pertaining to *profession*. Hence, among the words not recognized as a *profession* (i.e., B-PROF), the following stand out: *administartivo, anosPianista, auxilar, Empelada, hematolog, MA'p, medioc, trauamtologo*. As it can be appreciated, the model fails to recognise typos and misspellings.

6.6 Costs

Assuming a Tesla T4 GPU with a standard RAM environment, a 1.96 computation unit cost/hour (0.033/minute), a price of 0.102€/computation unit and a training time of 14 hours, each trained model costs 2.83€(without considering other costs, such as coding time or debugging time). The total cost per model can be calculated as follows:

$$TotalCost = 1.96 * \frac{51.12}{500} * (TraningTime) + 1.96 * \frac{51.12}{500} * [CodingTime + DebuggingTime]$$

In the previous mathematical expression, the codding time and the debugging time is fixed for all the models.

Chapter 7

Discussion, Conclusion and Future Perspectives

7.1 Discussion

Throughout the document, the importance of the occupation detection task has been highlighted. After reviewing the literature, it seems that this phenomenon is less explored than others that attempt to identify entities and modifiers within clinical narratives such as uncertainty, negation, and so on. However, the COVID-19 pandemic outbreak urged the need for systems capable of detecting professional groups at higher risk. Nevertheless, other applications and medical disciplines can take advantage of a system for occupation recognition. For example, in rheumatology, mechanical and inflammatory diseases such as tendinitis, or low back pain could be studied from an occupational perspective. Other specialities, such as pulmonology or oncology could benefit from having occupational information since patients could have been exposed to harmful substances in their workplace.

Usually, efforts on NER systems are focused on the English language. The most immediate consequence of this is that most of the resources, such as corpora, are limited to this language. In this work, we tried to identify occupation-related entities in Spanish clinical narratives. According to the last reports from Instituto Cervantes, Spanish is the fourth most spoken language in the world [171]. Therefore, the relevance of building Spanish NLP systems seem to be justified by these figures.

The objectives pursued in this Master's thesis are discussed below:

> **Objective 1**: To develop a system capable of detecting occupation mentions in clinical narratives

> To address this objective, we have employed different transformers' models based on BERT. The performance of these models is closely linked to the attention mechanism that allows the models to obtain contextual information. Most of the models used (e.g., BETO, ALBETO, DistilBETO) were originally pre-trained with general-domain corpus and then fine-tuned using the MEDDOPROF training set, however, one model was pre-trained specifically with biomedical and clinical data. The results obtained by this architecture outperformed the rest of the models. In addition, different design decisions and hyperparameter combinations were tested for all the models.
> With all of the above, we developed a NER based on transformers capable of identifying occupations in Spanish clinical and biomedical texts with a microaveraged F1 value of 0.664 in the test set.

Objective 2: To develop a system capable of detecting to whom the occupation mentions of objective 1 belong

The methodology applied to achieve this objective is similar to the one applied to objective 1. In this case, the best-performing architecture was also based on the model pre-trained with clinical and biomedical data. The model that achieved the best results has the same hyperparameters combination as the best-performing model of objective 1. However, the microaveraged F1 score obtained was significantly higher, 0.742.

Objective 3: Evaluation of the systems developed in objectives 1 and 2 with a collection of real clinical notes from the Hospital Clínico San Carlos (HCSC) Rheumatology Service

To achieve this objective, 2,000 clinical notes from the HCSC Rheumatology Service were used to evaluate the best-performing model of objectives 1 and 2. In this case, the model was trained with both training and validation sets. No fine-tuning was done for this objective.

Much of the work conducted in this Master's thesis was based on the hypothesis that by extending the training set with an additional corpus, the results of objectives 1 and 2 would improve. However, this did not happen. There are two different hypotheses that could explain this phenomenon.

- Bad annotation: MOD corpus was only annotated by one annotator, without prior experience, so the reliability of the annotation could not be evaluated.
- Non-informative training examples: since the notes that comprised the MOD corpus were obtained from a similar source to that of MEDDOPROF, the notes may not contribute with relevant and fresh information to the model. Moreover, it would have been interesting to incorporate annotations on the cases that posed the greatest challenge for the model to identify. Therefore, the MOD corpus could have been created after an initial error analysis.

It is important to note that there are no major differences between the characteristics of the two corpora (i.e., MEDDOPROF and MOD), as shown in Sections 4.1 and 4.2.5. Moreover, the notes comprising both corpora came from a common data source (i.e., TEMU-BSC).

On the other hand, the annotation process is a time-consuming task that requires experienced annotators related to the task field of study, such as linguists or physicians (which may result in increased costs); preferably, more than one, to assess the annotation agreement and to elaborate sufficiently descriptive annotation guidelines. We have shown that this task is prone to errors due to a) its manual nature and b) the particularities and intrinsic mechanisms of languages, with complex syntactic constructions. For example, duplicate annotations and duplicate notes were found in the MEDDOPROF corpus. Mechanisms to reduce this annotation burden have been proposed, such as the one followed in the MEDDOPROF task, a semi-supervised approach. In this approach, only 500 notes were manually annotated and the rest were automatically annotated and further reviewed by the annotation team. In this work, we decided to exclusively annotate manually, in order to have closer knowledge of the data to be worked with. This manual annotation required multiple readings of the annotation guidelines and a review process to assess the correctness of the annotations. Therefore, annotation guidelines are crucial for a NER recognition system to succeed. For instance, in Table 7.1, an example of why annotation guides are needed is shown. At first sight, this sentence could be annotated in three different ways depending on what is considered as *Sanitario* tag, and depending on whether overlapping entities are considered valid or not.

Table 7.1: Annotation example

	Es	atendido	por	personal	de	transporte	y	soporte	vital	avanzado
# 1	O	O	O	B-SAN	I-SAN	I-SAN	I-SAN	I-SAN	I-SAN	I-SAN
# 2	O	O	O	B-SAN	I-SAN	I-SAN	O	B-SAN	I-SAN	I-SAN
# 3	O	O	O	B-OTROS	I-OTROS	I-OTROS	O	B-SAN	I-SAN	I-SAN

B: Begin, I: Inside, O: Outside, SAN: Sanitario (health professional)

As we did not have a second annotator, agreement measures (i.e., consistency analysis) of this new annotated corpus, MOD, could not be established, although it would be desirable. The annotated corpus can be found in GitHub. Some complicated cases to annotate can be seen below.

- *Trabaja en la huerta / Trabaja de forma habitual en el campo / Trabaja en el cuidado de*: These tokens could be considered an activity or a proffesion. In some cases, there is not enough context to discriminate between professions and working status tags.

Some inconsistencies / unclear situations were found in the annotation guidelines.

- *Sick leave (baja laboral)* and *Work leave* (excedencia) concept was annotated, but not *labor discharge* (alta laboral).

- *Cursos de formación* was not annotated, but *Cursos de frigorista* and *aceptó asistir a cursos de formación* were annotated.

- Uncertainty between occupation and working status. For instance, pasaba tiempo trabajando was annotated as a profession, however, the working status tag could also fit with this mention.

Once again, the previous cases highlight the importance of the annotation guidelines and the importance of a trained annotator. To have a perfect control over what is expected to be recognised, the development of own annotation guides and own training corpus is desirable

Other relevant aspects that should be mentioned regarding the creation of the MOD corpus:

- The number of false positive notes when applying the rule-based algorithm with the gazetteer was high. In addition, the use of the gazetteer did not add relevant information. Therefore, instead of applying a gazetteer to identify the candidate notes for annotation, a preliminary transformer model trained with MEDDOPROF data could be launched on notes from the rest of the corpus and a manual review performed, mimicking an active learning approach.

- Considerable efforts were made to avoid data leakage when collecting additional clinical cases to enrich the training data. As the new clinical cases to annotate came from the same source (i.e., TEMU-BSC) as MEDDOPROF, notes extracted from other corpora, and therefore considered new at first sight, could be in the original MEDDOPROF training set, or even worse in the test set. By following a thorough processing pipeline consisting of duplicate removal and manual revision, this drawback was handled.

On the other hand, in this work, four different models were tested (i.e., BETO, RoBERTa, ALBERT, and DistilBERT), with a varying combination of hyperparameters and design decisions. This has allowed us to delve into different architectures and what works best for this task in general. However, there are an infinite number of possibilities that could have been considered when conducting this work, as shown by the different participant teams in the MEDDOPROF shared task. Nevertheless, with the experiments conducted, we have gained knowledge about how to work with transformers in a NER scenario.

Regarding objective 3, we have observed, as seen in Table 4.11, that in a real-world scenario, the prevalence of occupation-related information in patients' clinical notes is very low. Indeed, only

148 patient-profession entities (in 145 unique notes) appeared in a set of 2,000 notes, this is, a prevalence of 7.25%.

Finally, computational resources and complexity should not be overlooked. The models developed in this work are highly demanding on computational resources, making the use of GPUs mandatory. This was addressed using Google Colab Pro +, with an associated cost. This works because no GDPR-compliant data are used. However, in other scenarios, in which a pre-trained model should be fine-tuned with GDPR-compliant data, other alternatives, including training locally, should be considered.

7.2 Conclusions

Three main conclusions can be drawn from this work:

- The application of DL techniques based on transformers are useful in the recognition of named entities in EHR. Thanks to transfer learning and cloud computing frameworks it is not an indispensable requirement to have powerful workstations (as long as no personal data under GDPR regulation is used). Nowadays, training a transformer is not an overly complicated task, largely due to the effort of the academic community to generate documentation and tutorials with the aim of democratizing AI. Hugging Face is a good example of this.

- High-quality annotated data is required and almost mandatory to obtain reliable models. As we saw, adding the MOD corpus to the training data hinders the performance of the models.

- The clinical utility of large pre-trained models and fine-tuning is immeasurable given that a high proportion of the information stored in the EHR is unstructured and not all clinical centres have computational resources to train these models from scratch.

7.3 Dissemination activities

The work developed in this Master's thesis has been presented at the HCSC Rheumatology Unit and has been presented as an abstract, objective 3, at the *American College of Rheumatology Convergence 2023*. AbstractID: 1548841 (pending decision).

7.4 Future opportunities and research lines

The work carried out during this Master's thesis has laid the foundations for more detailed research into the identification of professions in EHR from the Rheumatology Service of the HCSC. The information extracted on occupation will be used to characterise different rheumatic and musculoskeletal patient populations. For instance, it is planned to measure the prevalence of this type of information in clinical notes and to study in which visits this type of information is collected, whether any population group is more likely to have this information, whether there is any difference in terms of the category of the practitioner treating the patient (attending, resident) in the collection of occupational information, and so on. It is also planned to study occupation in relation to patient diagnoses and comorbidities. Finally, this information is expected to be used in future research studies and to be included as independent variables in predictive models.

A noteworthy strength of contextual models is that they can identify occupations previously unseen. For example, "psicolologa", a typo of "psicóloga" (i.e., psychologist) is recognised as an occupation.

Leaving aside the most immediate application of these models in a real-world scenario such as the HCSC, throughout the development of this work many new models, libraries and frameworks have appeared, showing the interest in this field of AI by the different stakeholders. As an example, a library for automatic training and comparison of transformer models called NLPBOOST was published. Another library *designed for NLP researchers to easily utilise off-the-shelf algorithms and develop novel methods with user-defined models and tasks in real-world scenarios* called HugNLP

[172] has emerged. In addition, new and promising techniques for training transformers are being developed, such as using dual residual connections [173]. All these new developments could be taken into account in future iterations.

In addition, when annotating the MOD corpus, sentences such as *"Haber estado en contacto con uralita"* were found. The identification of agents such as air pollution, asthmagens, carcinogens, ergonomics, could be an interesting approach for future named entity recognition models.

Eventually, to improve the results of the models presented in this work, an additional annotator would be required to evaluate the MOD corpus and measure the inter-annotator agreement. Active learning approaches would be desirable.

7.5 Original contributions

As a result of the work carried out in this Master's thesis, the following outputs arose:

- Literature review of the occupation phenomenon as a SDOH.
- Study of the available Spanish corpus to enrich the MEDDOPROF training set with additional instances. Manual annotation of the occupation mentions of clinical cases coming from publicly available Spanish corpus was conducted to build the MOD corpus. This corpus is accessible through GitHub.
- Comparison of up to 40 models, pre-trained with general-domain and specific domain data, to address the occupation detection task and to whom the occupation belongs.
- Application of the best-performing model to a real-world scenario in which clinical notes from the HCSC Rheumatology Unit are used for occupation detection.

Appendix A

Appendix

Table A.1: Other resources facilitated by the MEDDOPROF shared task organiser team

Type	Name	Description
Evaluation Library	-	Official evaluation library
Linguistic Resources	CUTEXT	Medical term extraction tool
	SPACCC POS Tagger	Part Of Speech Tagger for Spanish medical domain corpus
	NegEx-MES	Spanish negation detection
	AbreMES-X	Generate Spanish Medical Abbreviation DataBase
	AbreMES-DB	Spanish Medical Abbreviation DataBase
	MeSpEn Glossaries	Bilingual medical glossaries
	Occupations gazetteer	Ocupations extracted from a set of terminologies (DeCS, ESCO, SnomedCT and WordNet) and Stanford CoreNLP
Word embeddings	FastText	Embeddings trained for medical Spanish domain
NLP Libraries	SpaCy	Python library
	NLTK	Python toolkit

A.1 BRAT deployment

BRAT installation guidelines can be found in the following link. Although the project has been discontinued, there is a strong community that offers support for BRAT newcommers. Therefore, assistance is available on the issue tracker at GitHub and at the BRAT google group. The working installation method followed in this Master's Thesis is:

- Install Git
- Install Python 3
- Clone the git repository

```
git clone https://github.com/nlplab/brat.git
```

- Install BRAT

```
cd brat
sudo chmod a+x install.sh
./install.sh
```

- Store the data in the *data* subdirectory
- Configure annotation tags, entities, relations, attributes, keyboard shortcuts, and other layout components by modifying *annotation.conf, kb_shortcuts.conf, tools.conf, visual.conf* files. The configuration guide can be found in the official webpage
- Configure the user and password credentials. This step is required in order to be able to annotate the data
- Launch BRAT (running the standalone server):

```
python3 standalone.py
```

- Go to the next URL in the browser http://0.0.0.0:8001

The configuration files (i.e., *annotation.conf*, *kb_shortcuts.conf*, *visual.conf*) used to annotate the clinical cases of MOD corpus can be found in GitHub.

A.2 Developed code

In this Appendix, the code files generated throughout this Master's Thesis are briefly described. In addition to Python Jupyter Notebooks (.ipynb), R has also been used (.R) [174]. All the scripts are reachable through GitHub and properly documented.

MOD processing pipeline:

1. *NotasARevisar.ipynb*: code that employs a rule-based algorithm and an occupation gazetteer to identify letters with potential occupation mentions for annotation. This code applies to BARR2, CANTEMIST, CodiEsp, LivingNER, MEDDOCAN, and PharmaCoNER corpus. The output of this code is a list of names of potential notes.
2. *ExtraccionNotas.ipynb*: script that searches the notes to annotate identified by *NotasARevisar.ipynb* in the different corpora directories, and copies them into a specified destination folder.
3. *DuplicadosNotas.ipynb*: code that searches for duplicate notes in both the notes identified for annotation, and the MEDDOPROF training and test set. Since there are duplicate notes with the same name, duplicate notes with different name and duplicate notes with slightly differences such as, identation level; different techniques are applied for the identification of such notes. Therefore, this code implements preprocessing steps such as converting to lowercase, removing special characters and stopwords, stemming and TF-IDF vectorisation to eventually perform document similarity analysis. Finally, the notes identified by this script are manually assessed and removed where appropriate.
4. *ProcesadoMOD.ipynb*: code that splits .ann files with multiple annotations into .ann files retaining only the tags of interest. Therefore, the input of this code are annotated notes with BRAT standoff format. In this work, this script was used to split each .ann file annotated with both task 1 and task 2 entities, into two files, one for each task. In addition, this code identifies the .txt files with at least one related annotation. By default, BRAT creates an .ann file for each .txt file even if the .txt is not annotated with an entity. With this script, the .txt files are filtered, and only those with at least one annotation in their corresponding .ann file are kept.
5. *ConversorBRATIOB.ipynb*: code to transform the .ann annotations into a suitable format that can be read and handled easily to construct the input to the neural network. Depending on the level of aggregation there are two options, aggregation at the clinical note level or at the sentence level. In both cases, *brat_to_conll* function from NeuroNER is used. This

script facilitates data conversion from .ann BRAT standoff format to BIO, using spaCy as a tokenizer. The output of this code is a single file with the annotations in BIO. Each sentence is separated from the rest by a blank line, allowing the addition of a sentence identifier. For this script to work, both *.ann* and *.txt* files should be in the same directory, and a tokenizer and a language should be specified as function parameters. This code is used interchangeably in MOD and MEDDOPROF notes.

6. *EstadisticasMOD.ipynb*: code that computes statistical measures to obtain insights from the MOD corpus, such as the distribution of entities; the minimum, maximum, average and total number of characters, tokens, entities and sentences per dataset. An equivalent code is built to obtain the MEDDOPROF corpus statistics (7. *EstadisticasMEDDOPROF.ipynb*).

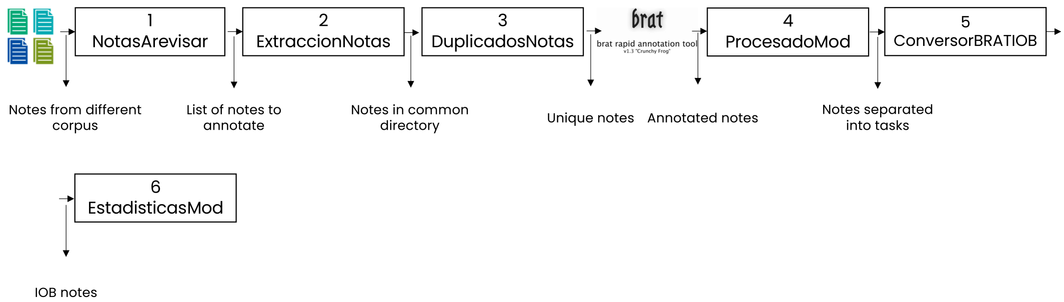

Figure A.1: Code pipeline for building MOD corpus

Main script:

8. *ModeloXXX Final.ipynb*: main document for training the transformers models, postprocessing and evaluation.

Error analysis:

9. *EvaluationLib.ipynb*: code to evaluate the best-performing model with scikit-learn, seqeval and nervaluate libraries.

10. *ErrorInspection.R*: code to study the tokens misclassified by the model.

A.3 Data access request

Originally, it was intended to train multilingual transformers models. Therefore, access to English language datasets that might contain information on occupations was requested and obtained.

A.4 Duplicate notes selection

HARVARD MEDICAL SCHOOL | BLAVATNIK INSTITUTE BIOMEDICAL INFORMATICS

Hi,

Your request to access the *n2c2 NLP Research Data Sets* dataset has been approved! Please click here to explore the dataset.

Thank you.

Figure A.2: n2c2 NLP Research Data Sets access confirmation

Dear Alfredo Madrid,

We are pleased to say that your "CITI Data or Specimens Only Research" training was approved.

You are now able to access protected databases upon agreeing to the terms of usage. For example, you can access MIMIC-III by following the steps below:

- Go to the project page at https://physionet.org/content/mimiciii/
- Find the “Files” section in the project description
- Click “Sign the data use agreement” to agree to the terms of usage for this dataset

Regards, The PhysioNet Team, MIT Laboratory for Computational Physiology Institute for Medical Engineering and Science, MIT, E25-505 77 Massachusetts Ave, Cambridge, MA 02139

Figure A.3: MIMIC-III data access confirmation

Table A.2: Duplicate note selection and removal from MOD corpus

Included notes	Included notes set	Excluded notes	Reason
S0210-56912007000200007-3	MOD	es-S0210-56912007000200007-3	E4: Same content different filename
S0376-78922009000300005-1	MOD	es-S0376-78922009000300005-1	E4: Same content different filename
S1137-66272013000200022-1	MOD	es-S1137-66272013000200022-1	E4: Same content different filename
S0210-56912009000900008-1	MOD	es-S0210-56912009000900008-1	E4: Same content different filename
S1137-66272006000100012-1	MOD	caso_clinico_radiologia867	E4: Same content different filename
cc_covid114	MOD	cc_covid81	E5: TF-IDF selected notes
casos_clinicos_cardiologia470	MOD	casos_clinicos_cardiologia44	E5: TF-IDF selected notes
cc_reumatologia240	MOD	cc_reumatologia238	E5: TF-IDF selected notes
casos_clinicos_cardiologia363	MOD	casos_clinicos_cardiologia187	E5: TF-IDF selected notes
casos_clinicos_cardiologia475	MOD	casos_clinicos_cardiologia47	E5: TF-IDF selected notes
casos_clinicos_cardiologia474	MOD	casos_clinicos_cardiologia46	E5: TF-IDF selected notes
casos_clinicos_cardiologia308	MOD	casos_clinicos_cardiologia165	E5: TF-IDF selected notes
casos_clinicos_profesiones132	MEDDO test	S0365-66912011001000003-4	E5: TF-IDF train+test+mod
casos_clinicos_profesiones79	MEDDO train	cc_reumatologia353	E5: TF-IDF train+test+mod
cc_reuma56	MEDDO test	cc_reumatologia60	E5: TF-IDF train+test+mod
casos_clinicos_profesiones3	MEDDO train	es-S0465-546X2014000400012-1	E5: TF-IDF train+test+mod
S1137-66272011000100013-1	MEDDO train	es-S1137-66272011000100013-1	E5: TF-IDF train+test+mod
S1137-66272011000100013-2	MEDDO train	es-S1137-66272011000100013-2	E5: TF-IDF train+test+mod
S1137-66272011000100013-3	MEDDO train	es-S1137-66272011000100013-3	E5: TF-IDF train+test+mod
caso_clinico_psiquiatria306	MEDDO train	S0211-57352014000400011-1	E5: TF-IDF train+test+mod
S0465-546X2014000300010-1	MEDDO train	es-S0465-546X2014000300010-1	E5: TF-IDF train+test+mod
cc_reuma58	MEDDO test	cc_reumatologia62	E5: TF-IDF train+test+mod
caso_clinico_psiquiatria305	MEDDO test	S0211-57352014000400010-1	E5: TF-IDF train+test+mod
caso_clinico_psiquiatria372	MEDDO train	cc_geneticas200	E5: TF-IDF train+test+mod
S0465-546X2009000300008-1	MEDDO train	es-S0465-546X2009000300008-1	E5: TF-IDF train+test+mod
S1137-66272014000100021-1	MEDDO train	es-S1137-66272014000100021-1	E5: TF-IDF train+test+mod
caso_clinico_psiquiatria278	MEDDO train	S0211-57352015000100011-2	E5: TF-IDF train+test+mod
S1132-62552015000100006-1	MEDDO train	es-S1132-62552015000100006-1	E5: TF-IDF train+test+mod
casos_clinicos_profesiones163	MEDDO test	S1578-25492016000400004-1	E5: TF-IDF train+test+mod
S0465-546X2011000300007-1	MEDDO train	es-S0465-546X2011000300007-1	E5: TF-IDF train+test+mod
S0376-78922009000100011-1	MEDDO test	es-S0376-78922009000100011-1.txt	E5: TF-IDF train+test+mod
casos_clinicos_profesiones228	MEDDO test	S0211-57352002000100009-1	E5: TF-IDF train+test+mod
casos_clinicos_profesiones1	MEDDO train	casos_clinicos_cardiologia377	E5: TF-IDF train+test+mod
caso_clinico_psiquiatria285	MEDDO train	S0211-57352014000300007-1	E5: TF-IDF train+test+mod
casos_clinicos_cardiologia335	MOD	casos_clinicos_cardiologia175	Manual review
casos_clinico_psiquiatria293	MEDDO train	S0211-57352013000300012-1	Manual review
casos_clinico_psiquiatria294	MEDDO train	S0211-57352013000400004-1	Manual review

MOD: More Occupation data, MEDDO: MEDDOPROF, TF-IDF: Term frequency – Inverse Document Frequency, E4: Exclusion criteria 4, E5: Exclusion criteria 5, TF-IDF selected noted means that notes similarity was assessed considering only the MOD corpus. E5: TF-IDF train+test+mod means that notes similarity was compared between the MOD corpus and the train and test sets from MEDDOPROF to ensure that no data leakage occurs

A.5 Special tokens

Table A.3: BERT special tokens.

Special Token	Token ID	Description
[PAD]	0	Used to pad variable-length sequences to the same length within a batch of input data.
[UNK]	100	Used to represent out-of-vocabulary (OOV) words during both training and inference when the model encounters a word that it hasn't seen before.
[CLS]	101	Marks the beginning of a sequence and is used as a classification token.
[SEP]	102	Marks the end of a sentence or a sequence. It is also used to separate pairs of sentences in sequence classification tasks.
[MASK]	103	Used to replace a word during pre-training with a probability of 15%. This is done to train the model to fill in missing words.

A.6 Special characters

Special symbols that must be taken into account when post-processing the predictions are shown in Figure A.4

!	;	0xad	—
"	<	®	“
%	=	°	”
'	>	·	•
(	?	»	…
)	[	½	→
*	]	¾	−
+	^	¿	∼
,	~	×	≤
-	0x8a	ß	≥
.	¡	ó	▪
/	§	µ	0xfeff
:	«	–	

Figure A.4: Symbols found in the test set clinical notes

A.7 BERT architectures comparison

Table A.4: Comparison of ALBERT, BERT, DistilBERT, RoBERTa

Transformer	Pre-training	Architecture	Parameters	Data Characteristics	Performance	Other
ALBERT [175]	MLM with SOP	Encoder (12/24 layers)	12M/18M/60M/235M	BookCorpus English Wikipedia	Achieves state-of-the-art performance with fewer parameters than BERT	Factorized Embedding Parameterization Cross-Layer Parameter Sharing Inter-sentence coherence loss
BERT [75]	MLM + NSP	Encoder (12/24 layers)	110M/340M	BookCorpus English Wikipedia	Baseline model	
DistilBERT [87]	MLM	Encoder (6 layers)	66M	BookCorpus English Wikipedia	Smaller and faster than BERT while retaining similar performance	Distillation Cosine-distance losses
RoBERTa [76]	MLM with dynamic masking	Encoder (12/24 layers)	125/355M	BookCorpus English Wikipedia CC News OpenWebText Stories	Outperforms BERT on many NLP tasks	Dynamic masking

MLM: Masked Language Model, NSP: Next Sentence Prediction, SOP: Sentence Order Prediction

A.8 Evaluation nervaluate

Hereafter, the evaluation according to the nervaluate library is shown. With this library, the authors intended to provide additional information that goes beyond the traditional evaluation schemas. To this end, they defined five metrics, Table A.5.

Table A.5: Metrics presented in nervaluate library. Source: https://github.com/MantisAI/nervaluate

Error type	Explanation
Correct (COR)	both, gold and prediction, are the same
Incorrect (INC)	the output of a system and the golden annotation don't match
Partial (PAR)	system and the golden annotation are somewhat "similar" but not the same
Missing (MIS)	a golden annotation is not captured by a system
Spurius (SPU)	system produces a response which doesn't exist in the golden annotation

They also established different ways to measure such metrics, Table A.6.

Table A.6: Measurement system presented in nervaluate library. Source: https://github.com/MantisAI/nervaluate

Evaluation schema	Explanation
Strict	exact boundary surface string match and entity type
Exact	exact boundary match over the surface string, regardless of the type
Partial	partial boundary match over the surface string, regardless of the type
Type	some overlap between the system tagged entity and the gold annotation is required
Spurius (SPU)	system produces a response which doesn't exist in the golden annotation

The following concepts were defined:

$$\begin{aligned}\text{POSSIBLE (POS)} &= \text{COR} + \text{INC} + \text{PAR} + \text{MIS} = \text{TP} + \text{FN}\\ \text{ACTUAL (ACT)} &= \text{COR} + \text{INC} + \text{PAR} + \text{SPU} = \text{TP} + \text{FP}\end{aligned}$$

And the precision / recall metrics were calculated as follows:

- Exact match (i.e., strict and exact):

$$\begin{aligned}\text{Precision} &= (\text{COR}/\text{ACT}) = \text{TP}/(\text{TP} + \text{FP})\\ \text{Recall} &= (\text{COR}/\text{POS}) = \text{TP}/(\text{TP} + \text{FN})\end{aligned}$$

- Partial match (i.e., partial and type):

$$\begin{aligned}&\text{Precision} = (\text{COR} + 0.5 \times \text{PAR})/\text{ACT} = \text{TP}/(\text{TP} + \text{FP})\\ &\text{Recall} = (\text{COR} + 0.5 \times \text{PAR})/\text{POS} = \text{COR}/\text{ACT} = \text{TP}/(\text{TP} + \text{FN})\end{aligned}$$

The strict column is similar to the values obtained in Table 6.5 and Table 6.5.

Table A.7: TASK1-NER results according to nervaluate library

Entity	Measure	Type	Partial	Strict	Exact
PROFESIÓN (PROFESSION)	Correct	642	600	593	600
	Incorrect	12	0	61	54
	Partial	0	54	0	0
	Missed	42	42	42	42
	Spurious	28	28	28	28
	Possible	696	696	696	696
	Actual	682	682	682	682
	Precision	0.94	0.92	0.87	0.88
	Recall	0.92	0.90	0.85	0.86
	F1	0.93	0.91	0.86	0.87
SITUACIÓN LABORAL (WORKING STATUS)	Correct	279	243	236	243
	Incorrect	8	0	51	44
	Partial	0	44	0	0
	Missed	70	70	70	70
	Spurious	59	59	59	59
	Possible	357	357	357	357
	Actual	346	346	346	346
	Precision	0.81	0.77	0.68	0.70
	Recall	0.78	0.74	0.66	0.68
	F1	0.79	0.75	0.67	0.69
ACTIVIDAD (ACTIVITY)	Correct	14	15	10	15
	Incorrect	7	0	11	6
	Partial	0	6	0	0
	Missed	7	7	7	7
	Spurious	16	16	16	16
	Possible	28	28	28	28
	Actual	37	37	37	37
	Precision	0.38	0.49	0.27	0.41
	Recall	0.5	0.64	0.36	0.54
	F1	0.43	0.55	0.31	0.46
Total	Correct	935	858	839	858
	Incorrect	27	0	123	104
	Partial	0	104	0	0
	Missed	119	119	119	119
	Spurious	103	103	103	103
	Possible	1081	1081	1081	1081
	Actual	1065	1065	1065	1065
	Precision	0.88	0.85	0.79	0.80
	Recall	0.86	0.84	0.78	0.79
	F1	0.87	0.85	0.78	0.80

Table A.8: TASK2-CLASS results according to nervaluate library

Entity	Measure	Type	Partial	Strict	Exact
FAMILIAR (FAMILY MEMBER)	Correct	35	38	31	38
	Incorrect	9	0	13	6
	Partial	0	6	0	0
	Missed	8	8	8	8
	Spurious	11	11	11	11
	Possible	52	52	52	52
	Actual	55	55	55	55
	Precision	0.64	0.75	0.56	0.69
	Recall	0.67	0.79	0.60	0.73
	F1	0.65	0.77	0.58	0.71
SANITARIO (HEALTH PROFESSIONAL)	Correct	286	278	276	278
	Incorrect	2	0	12	10
	Partial	0	10	0	0
	Missed	5	5	5	5
	Spurious	5	5	5	5
	Possible	293	293	293	293
	Actual	293	293	293	293
	Precision	0.98	0.97	0.94	0.95
	Recall	0.98	0.97	0.94	0.95
	F1	0.98	0.97	0.94	0.95
OTROS (OTHER)	Correct	100	114	96	114
	Incorrect	20	0	24	6
	Partial	0	6	0	0
	Missed	26	26	26	26
	Spurious	19	19	19	19
	Possible	146	146	146	146
	Actual	139	139	139	139
	Precision	0.72	0.84	0.69	0.82
	Recall	0.68	0.80	0.66	0.78
	F1	0.70	0.82	0.67	0.80
PACIENTE (PATIENT)	Correct	473	417	400	417
	Incorrect	20	0	93	76
	Partial	0	76	0	0
	Missed	97	97	97	97
	Spurious	88	88	88	88
	Possible	590	590	590	590
	Actual	581	581	581	581
	Precision	0.81	0.78	0.69	0.72
	Recall	0.80	0.77	0.68	0.71
	F1	0.81	0.78	0.68	0.71
Total	Correct	894	847	803	847
	Incorrect	51	0	142	98
	Partial	0	98	0	0
	Missed	136	136	136	136
	Spurious	123	123	123	123
	Possible	1081	1081	1081	1081
	Actual	1068	1068	1068	1068
	Precision	0.84	0.84	0.75	0.79
	Recall	0.83	0.83	0.74	0.78
	F1	0.83	0.83	0.75	0.79

Bibliography

[1] World Health Organization and International Labour Organization. *WHO/ILO joint estimates of the work-related burden of disease and injury, 2000-2016: global monitoring report.* World Health Organization, 2021.

[2] David H. Wegman, Catharyn T. Liverman, Andrea M. Schultz, et al. *Incorporating occupational information in electronic health records: Letter report.* National Academies Press, Oct. 2011, pp. 1–74. ISBN: 0309217431. DOI: 10.17226/13207.

[3] *National Occupational Research Agenda (96-115).* June 2014. URL: https://www.cdc.gov/niosh/docs/96-115/default.html.

[4] *Work-related diseases.* Accessed: 2023-02-03. URL: https://osha.europa.eu/en/themes/work-related-diseases.

[5] Ana Llena-Nozal, Maarten Lindeboom, and France Portrait. "The effect of work on mental health: does occupation matter?" In: *Health Economics* 13.10 (2004), pp. 1045–1062. DOI: https://doi.org/10.1002/hec.929. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/hec.929. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/hec.929.

[6] Michele Belloni, Ludovico Carrino, and Elena Meschi. "The impact of working conditions on mental health: Novel evidence from the UK". In: *Labour Economics* 76 (2022), p. 102176. ISSN: 0927-5371. DOI: https://doi.org/10.1016/j.labeco.2022.102176. URL: https://www.sciencedirect.com/science/article/pii/S0927537122000677.

[7] Rajeswari Sambasivam, Anitha Jeyagurunathan, Edimansyah Abdin, et al. "Occupational groups and its physical and mental health correlates: results from the Singapore Mental Health Study 2016". In: *International Archives of Occupational and Environmental Health* 95 (3 Apr. 2022), pp. 753–764. ISSN: 0340-0131. DOI: 10.1007/s00420-021-01741-8.

[8] Bastian Ravesteijn, Hans van Kippersluis, and Eddy van Doorslaer. *The Contribution of Occupation to Health Inequality.* Dec. 2013, pp. 311–332. DOI: 10.1108/S1049-2585(2013)0000021014.

[9] Ralitza Gueorguieva, Jody L. Sindelar, Tracy A. Falba, et al. "The Impact of Occupation on Self-Rated Health: Cross-Sectional and Longitudinal Evidence from the Health and Retirement Survey". In: *The Journals of Gerontology: Series B* 64B.1 (Feb. 2009), pp. 118–124. ISSN: 1079-5014. DOI: 10.1093/geronb/gbn006. eprint: https://academic.oup.com/psychsocgerontology/article-pdf/64B/1/118/1699897/gbn006.pdf. URL: https://doi.org/10.1093/geronb/gbn006.

[10] Egidio Riva, Mario Lucchini, and Carlotta Piazzoni. "The effect of job quality on quality of life and wellbeing in later career stages: A multilevel and longitudinal analysis on older workers in Europe". In: *Applied Research in Quality of Life* 17 (4 Aug. 2022), pp. 1993–2015. ISSN: 1871-2584. DOI: 10.1007/s11482-021-10021-z.

[11] O.P. Steeno and A. Pangkahila. "Occupational Influences on Male Fertility and Sexuality". In: *Andrologia* 16.1 (1984), pp. 5–22. DOI: https://doi.org/10.1111/j.1439-0272.1984.tb00227.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1439-0272.1984.tb00227.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1439-0272.1984.tb00227.x.

[12] Matthew Schmitz and Linda Forst. "Industry and Occupation in the Electronic Health Record: An Investigation of the National Institute for Occupational Safety and Health Industry and Occupation Computerized Coding System". In: *JMIR Med Inform* 4.1 (Feb. 2016), e5. ISSN: 2291-9694. DOI: 10.2196/medinform.4839. URL: http://www.ncbi.nlm.nih.gov/pubmed/26878932.

[13] Víctor Suárez-Paniagua and Arlene Casey. "BERT and Approximate String Matching for Automatic Recognition and Normalization of Professions in Spanish Medical Documents." In: *IberLEF@ SEPLN*. 2021, pp. 803–813.

[14] World Health Organization. *A conceptual framework for action on the social determinants of health*. 2010.

[15] Stacey Marovich, Genevieve Barkocy Luensman, Barbara Wallace, et al. "Opportunities at the intersection of work and health: Developing the occupational data for health information model". In: *Journal of the American Medical Informatics Association* 27.7 (June 2020), pp. 1072–1083. ISSN: 1527-974X. DOI: 10.1093/jamia/ocaa070. eprint: https://academic.oup.com/jamia/article-pdf/27/7/1072/34152921/ocaa070.pdf. URL: https://doi.org/10.1093/jamia/ocaa070.

[16] Natasha Chilman, Xingyi Song, Angus Roberts, et al. "Text mining occupations from the mental health electronic health record: a natural language processing approach using records from the Clinical Record Interactive Search (CRIS) platform in south London, UK". In: *BMJ Open* 11.3 (2021). ISSN: 2044-6055. DOI: 10.1136/bmjopen-2020-042274. eprint: https://bmjopen.bmj.com/content/11/3/e042274.full.pdf. URL: https://bmjopen.bmj.com/content/11/3/e042274.

[17] Ranyah Aldekhyyel, Elizabeth S Chen, Sripriya Rajamani, et al. "Content and Quality of Free-Text Occupation Documentation in the Electronic Health Record." In: *AMIA ... Annual Symposium proceedings. AMIA Symposium* 2016 (2016), pp. 1708–1716. ISSN: 1942-597X.

[18] Michela Assale, Linda Greta Dui, Andrea Cina, et al. "The Revival of the Notes Field: Leveraging the Unstructured Content in Electronic Health Records". In: *Frontiers in Medicine* 6 (Apr. 2019). ISSN: 2296-858X. DOI: 10.3389/fmed.2019.00066.

[19] Stephen Wu, Kirk Roberts, Surabhi Datta, et al. "Deep learning in clinical natural language processing: a methodical review". In: *Journal of the American Medical Informatics Association* 27 (3 Mar. 2020), pp. 457–470. ISSN: 1527-974X. DOI: 10.1093/jamia/ocz200.

[20] Lemai Nguyen, Emilia Bellucci, and Linh Thuy Nguyen. "Electronic health records implementation: An evaluation of information system impact and contingency factors". In: *International Journal of Medical Informatics* 83 (11 Nov. 2014), pp. 779–796. ISSN: 13865056. DOI: 10.1016/j.ijmedinf.2014.06.011.

[21] Jun Liang, Ying Li, Zhongan Zhang, et al. "Adoption of Electronic Health Records (EHRs) in China During the Past 10 Years: Consecutive Survey Data Analysis and Comparison of Sino-American Challenges and Experiences". In: *Journal of Medical Internet Research* 23 (2 Feb. 2021), e24813. ISSN: 1438-8871. DOI: 10.2196/24813.

[22] Kory Kreimeyer, Matthew Foster, Abhishek Pandey, et al. "Natural language processing systems for capturing and standardizing unstructured clinical information: A systematic review". In: *Journal of Biomedical Informatics* 73 (2017), pp. 14–29. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2017.07.012. URL: https://www.sciencedirect.com/science/article/pii/S1532046417301685.

[23] Zexian Zeng, Yu Deng, Xiaoyu Li, et al. "Natural Language Processing for EHR-Based Computational Phenotyping". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 16.1 (2019), pp. 139–153. DOI: 10.1109/TCBB.2018.2849968.

[24] Hercules Dalianis. *Clinical Text Mining*. Springer International Publishing, 2018. ISBN: 978-3-319-78502-8. DOI: 10.1007/978-3-319-78503-5.

[25] Wei-Qi Wei, Pedro L Teixeira, Huan Mo, et al. "Combining billing codes, clinical notes, and medications from electronic health records provides superior phenotyping performance". In: *Journal of the American Medical Informatics Association* 23 (e1 Apr. 2016), e20–e27. ISSN: 1527-974X. DOI: 10.1093/jamia/ocv130.

[26] Milena A. Gianfrancesco and Neal D. Goldstein. "A narrative review on the validity of electronic health record-based research in epidemiology". In: *BMC Medical Research Methodology* 21 (1 Dec. 2021), p. 234. ISSN: 1471-2288. DOI: 10.1186/s12874-021-01416-5.

[27] Thomas Searle, Zina Ibrahim, James Teo, et al. "Estimating redundancy in clinical text". In: *Journal of Biomedical Informatics* 124 (2021), p. 103938. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2021.103938. URL: https://www.sciencedirect.com/science/article/pii/S1532046421002677.

[28] Elizabeth A Lindemann, Elizabeth S Chen, Sripriya Rajamani, et al. "Assessing the representation of occupation information in free-text clinical documents across multiple sources". In: *Studies in health technology and informatics* 245 (2017), p. 486.

[29] Jimmy Phuong, Elizabeth Zampino, Nicholas Dobbins, et al. "Extracting Patient-level Social Determinants of Health into the OMOP Common Data Model". In: *AMIA Annual Symposium Proceedings*. Vol. 2021. American Medical Informatics Association. 2021, p. 989.

[30] Alfredo Madrid García. "Data-driven approaches in rheumatology: learning from real-world data". Nov. 2022. DOI: 10.20868/UPM.thesis.72300. URL: https://oa.upm.es/72300/.

[31] Azad Dehghan, Tom Liptrot, Daniel Tibble, et al. "Identification of Occupation Mentions in Clinical Narratives". In: *Natural Language Processing and Information Systems*. Ed. by Elisabeth Métais, Farid Meziane, Mohamad Saraee, et al. Cham: Springer International Publishing, 2016, pp. 359–365. ISBN: 978-3-319-41754-7.

[32] Mehmet Kayaalp, Allen C Browne, Pamela Sagan, et al. "Challenges and insights in using HIPAA privacy rule for clinical text annotation". In: *AMIA Annual Symposium proceedings*. Vol. 2015. American Medical Informatics Association. 2015, p. 707.

[33] Richárd Farkas, Veronika Vincze, György Móra, et al. "The CoNLL-2010 Shared Task: Learning to Detect Hedges and their Scope in Natural Language Text". In: *Proceedings of the Fourteenth Conference on Computational Natural Language Learning – Shared Task*. Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 1–12. URL: https://aclanthology.org/W10-3001.

[34] Roser Morante and Eduardo Blanco. "* sem 2012 shared task: Resolving the scope and focus of negation". In: ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. 2012, pp. 265–274.

[35] Majid Rastegar-Mojarad, Sijia Liu, Yanshan Wang, et al. "BioCreative/OHNLP Challenge 2018". In: BCB '18. Washington, DC, USA: Association for Computing Machinery, 2018, p. 575. ISBN: 9781450357944. DOI: 10.1145/3233547.3233672. URL: https://doi.org/10.1145/3233547.3233672.

[36] Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. "Evaluating temporal relations in clinical text: 2012 i2b2 Challenge". In: *Journal of the American Medical Informatics Association* 20.5 (Apr. 2013), pp. 806–813. ISSN: 1067-5027. DOI: 10.1136/amiajnl-2013-001628. eprint: https://academic.oup.com/jamia/article-pdf/20/5/806/17374624/20-5-806.pdf. URL: https://doi.org/10.1136/amiajnl-2013-001628.

[37] Salvador Lima-López, Eulàlia Farré-Maduell, Antonio Miranda-Escalada, et al. "NLP applied to occupational health: MEDDOPROF shared task at IberLEF 2021 on automatic recognition, classification and normalization of professions and occupations from medical texts". In: *Procesamiento del Lenguaje Natural* 67 (2021), pp. 243–256. ISSN: 1989-7553. URL: http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6393.

[38] Antonio Miranda-Escalada, Eulàlia Farré-Maduell, Salvador Lima-López, et al. "The ProfNER shared task on automatic recognition of occupation mentions in social media: systems, evaluation, guidelines, embeddings and corpora". In: *Proceedings of the Sixth Social Media Mining for Health (#SMM4H) Workshop and Shared Task*. Mexico City, Mexico: Association for Computational Linguistics, June 2021, pp. 13–20. DOI: 10.18653/v1/2021.smm4h-1.3. URL: https://aclanthology.org/2021.smm4h-1.3.

[39] *Call for participation: MEDDOPROF shared task: Medical Documents Profession Recognition Shared Task (Iberlef - SEPLN 2021)*. URL: https://inb-elixir.es/events/call-participation-meddoprof-shared-task-medical-documents-profession-recognition-shared.

[40] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.", 2022.

[41] Ashish Bansal. *Advanced Natural Language Processing with TensorFlow 2: Build effective real-world NLP applications using NER, RNNs, seq2seq models, Transformers, and more*. Packt Publishing Ltd, 2021.

[42] Sudharsan Ravichandiran. *Getting Started with Google BERT: Build and train state-of-the-art natural language processing models using BERT*. Packt Publishing Ltd, 2021.

[43] Shashank Mohan Jain. "Introduction to Transformers". In: *Introduction to Transformers for NLP: With the Hugging Face Library and Models to Solve Problems*. Berkeley, CA: Apress, 2022, pp. 19–36. ISBN: 978-1-4842-8844-3. DOI: 10.1007/978-1-4842-8844-3_2. URL: https://doi.org/10.1007/978-1-4842-8844-3_2.

[44] Savas Yildirim and Meysam Asgari-Chenaghlu. *Mastering Transformers: Build state-of-the-art models from scratch with advanced natural language processing techniques*. Packt Publishing Ltd, 2021.

[45] Akshay Kulkarni, Adarsha Shivananda, and Anoosh Kulkarni. *Natural Language Processing Projects*. Springer, 2022. DOI: https://doi.org/10.1007/978-1-4842-7386-9.

[46] Lewis Tunstall, Leandro von Werra, and Thomas Wolf. *Natural language processing with transformers*. " O'Reilly Media, Inc.", 2022.

[47] Dan Jurafsky. *Speech & language processing 3rd edition draft*. 2021.

[48] Tadej. Magajna. *Natural language processing with Flair a practical guide to understanding and solving NLP problems with Flair*. Packt Publishing, 2022. ISBN: 9781801072311.

[49] Ankur A Patel and Ajay Uppili Arasanipalai. *Applied Natural Language Processing in the Enterprise*. " O'Reilly Media, Inc.", 2021.

[50] Eli Stevens, Luca Antiga, and Thomas Viehmann. *Deep learning with PyTorch*. Manning Publications, 2020.

[51] Masato Hagiwara. *Real-World Natural Language Processing: Practical Applications with Deep Learning*. Simon and Schuster, 2021.

[52] Paul Azunre. *Transfer learning for natural language processing*. Simon and Schuster, 2021.

[53] Denis Rothman. *Transformers for Natural Language Processing: Build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more*. Packt Publishing Ltd, 2021.

[54] *Hugging face – the AI community building the future*. Accessed: 2023-02-03. URL: https://huggingface.co/.

[55] *Named-entity recognition*. Accessed: October, 2022. Oct. 2022. URL: https://en.wikipedia.org/wiki/Named-entity_recognition.

[56] Nadeesha Perera, Matthias Dehmer, and Frank Emmert-Streib. "Named Entity Recognition and Relation Detection for Biomedical Information Extraction". In: *Frontiers in Cell and Developmental Biology* 8 (Aug. 2020). ISSN: 2296-634X. DOI: 10.3389/fcell.2020.00673.

[57] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[58] GuoDong Zhou, Jie Zhang, Jian Su, et al. "Recognizing names in biomedical texts: a machine learning approach". In: *Bioinformatics* 20.7 (Feb. 2004), pp. 1178–1190. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bth060. eprint: https://academic.oup.com/bioinformatics/article-pdf/20/7/1178/679155/bth060.pdf. URL: https://doi.org/10.1093/bioinformatics/bth060.

[59] Ulf Leser and Jörg Hakenberg. "What makes a gene name? Named entity recognition in the biomedical literature". In: *Briefings in Bioinformatics* 6.4 (Dec. 2005), pp. 357–369. ISSN: 1467-5463. DOI: 10.1093/bib/6.4.357. eprint: https://academic.oup.com/bib/article-pdf/6/4/357/9731832/357.pdf. URL: https://doi.org/10.1093/bib/6.4.357.

[60] Hye-Jeong Song, Byeong-Cheol Jo, Chan-Young Park, et al. "Comparison of named entity recognition methodologies in biomedical documents". In: *BioMedical Engineering OnLine* 17 (S2 Nov. 2018), p. 158. ISSN: 1475-925X. DOI: 10.1186/s12938-018-0573-6.

[61] Gökberk Çelikmasat, Muhammed Enes Aktürk, Yunus Emre Ertunç, et al. "Biomedical Named Entity Recognition Using Transformers with biLSTM + CRF and Graph Convolutional Neural Networks". In: *2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. 2022, pp. 1–6. DOI: 10.1109/INISTA55318.2022.9894270.

[62] Guillermo Moncecchi, Jean-Luc Minel, and Dina Wonsever. "Improving Speculative Language Detection using Linguistic Knowledge". In: *Proceedings of the Workshop on Extra-Propositional Aspects of Meaning in Computational Linguistics*. Jeju, Republic of Korea: Association for Computational Linguistics, July 2012, pp. 37–46. URL: https://aclanthology.org/W12-3805.

[63] Lance Ramshaw and Mitch Marcus. "Text Chunking using Transformation-Based Learning". In: *Third Workshop on Very Large Corpora*. 1995. URL: https://aclanthology.org/W95-0107.

[64] Han-Cheol Cho, Naoaki Okazaki, Makoto Miwa, et al. "Named entity recognition with multiple segment representations". In: *Information Processing & Management* 49.4 (2013), pp. 954–965. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2013.03.002. URL: https://www.sciencedirect.com/science/article/pii/S0306457313000368.

[65] Shaina Raza, Elham Dolatabadi, Nancy Ondrusek, et al. "Discovering Social Determinants of Health from Case Reports using Natural Language Processing: Algorithmic Development and Validation". In: *medRxiv* (2022). DOI: 10.1101/2022.11.30.22282946. eprint: https://www.medrxiv.org/content/early/2022/12/06/2022.11.30.22282946.full.pdf. URL: https://www.medrxiv.org/content/early/2022/12/06/2022.11.30.22282946.

[66] Yukun Chen, Thomas A. Lasko, Qiaozhu Mei, et al. "A study of active learning methods for named entity recognition in clinical text". In: *Journal of Biomedical Informatics* 58 (2015), pp. 11–18. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2015.09.010. URL: https://www.sciencedirect.com/science/article/pii/S1532046415002038.

[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[68] Jordi de la Torre. *Transformadores: Fundamentos teoricos y Aplicaciones*. 2023. arXiv: 2302.09327 [cs.CL].

[69] Caroline Becker. *NLP Seminar at LMU Munich*. https://slds-lmu.github.io/seminar_nlp_ss20/. Accessed: March 26, 2023. 2020.

[70] Jeremy Howard and Sebastian Ruder. *Universal Language Model Fine-tuning for Text Classification*. 2018. DOI: 10.48550/ARXIV.1801.06146. URL: https://arxiv.org/abs/1801.06146.

[71] Zihan Liu, Yan Xu, Tiezheng Yu, et al. *CrossNER: Evaluating Cross-Domain Named Entity Recognition*. 2020. arXiv: 2012.04373 [cs.CL].

[72] Tianyang Lin, Yuxin Wang, Xiangyang Liu, et al. "A Survey of Transformers". In: *CoRR* abs/2106.04554 (2021). arXiv: 2106.04554. URL: https://arxiv.org/abs/2106.04554.

[73] Xavier Amatriain. *Transformer models: an introduction and catalog*. 2023. DOI: 10.48550/ARXIV.2302.07730. URL: https://arxiv.org/abs/2302.07730.

[74] Chengwei Wei, Yun-Cheng Wang, Bin Wang, et al. *An Overview on Language Models: Recent Developments and Outlook*. 2023. DOI: 10.48550/ARXIV.2303.05759. URL: https://arxiv.org/abs/2303.05759.

[75] Jacob Devlin, Ming-Wei Chang, Kenton Lee, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[76] Yinhan Liu, Myle Ott, Naman Goyal, et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. DOI: 10.48550/ARXIV.1907.11692. URL: https://arxiv.org/abs/1907.11692.

[77] Chi Sun, Xipeng Qiu, Yige Xu, et al. *How to Fine-Tune BERT for Text Classification?* 2019. DOI: 10.48550/ARXIV.1905.05583. URL: https://arxiv.org/abs/1905.05583.

[78] Iz Beltagy, Matthew E. Peters, and Arman Cohan. *Longformer: The Long-Document Transformer*. 2020. DOI: 10.48550/ARXIV.2004.05150. URL: https://arxiv.org/abs/2004.05150.

[79] Raghavendra Pappagari, Piotr Żelasko, Jesús Villalba, et al. *Hierarchical Transformers for Long Document Classification*. 2019. arXiv: 1910.10781 [cs.CL].

[80] Ashutosh Adhikari, Achyudh Ram, Raphael Tang, et al. *DocBERT: BERT for Document Classification*. 2019. arXiv: 1904.08398 [cs.CL].

[81] Ruixuan Zhang, Zhuoyu Wei, Yu Shi, et al. *{BERT}-{AL}: {BERT} for Arbitrarily Long Document Understanding*. 2020. URL: https://openreview.net/forum?id=SklnVAEFDB.

[82] Yikuan Li, Ramsey M Wehbe, Faraz S Ahmad, et al. "A comparative study of pretrained language models for long clinical text". In: *Journal of the American Medical Informatics Association* 30.2 (Nov. 2022), pp. 340–347. ISSN: 1527-974X. DOI: 10.1093/jamia/ocac225. eprint: https://academic.oup.com/jamia/article-pdf/30/2/340/48754733/ocac225.pdf. URL: https://doi.org/10.1093/jamia/ocac225.

[83] José Cañete. *Compilation of Large Spanish Unannotated Corpora*. Zenodo, May 2019. DOI: 10.5281/zenodo.3247731. URL: https://doi.org/10.5281/zenodo.3247731.

[84] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, et al. "BioBERT: A pre-trained biomedical language representation model for biomedical text mining". In: *Bioinformatics* 36 (4 Feb. 2020), pp. 1234–1240. ISSN: 14602059. DOI: 10.1093/bioinformatics/btz682.

[85] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. "ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission". In: *CoRR* abs/1904.05342 (2019). arXiv: 1904.05342. URL: http://arxiv.org/abs/1904.05342.

[86] Emily Alsentzer, John R. Murphy, Willie Boag, et al. "Publicly Available Clinical BERT Embeddings". In: *CoRR* abs/1904.03323 (2019). arXiv: 1904.03323. URL: http://arxiv.org/abs/1904.03323.

[87] Victor Sanh, Lysandre Debut, Julien Chaumond, et al. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. 2019. DOI: 10.48550/ARXIV.1910.01108. URL: https://arxiv.org/abs/1910.01108.

[88] Casimiro Pio Carrino, Jordi Armengol-Estapé, Asier Gutiérrez-Fandiño, et al. *Biomedical and Clinical Language Models for Spanish: On the Benefits of Domain-Specific Pretraining in a Mid-Resource Scenario*. 2021. arXiv: 2109.03570 [cs.CL].

[89] Lukas Lange, Heike Adel, Jannik Strötgen, et al. "CLIN-X: pre-trained language models and a study on cross-task transfer for concept extraction in the clinical domain". In: *Bioinformatics* 38.12 (Apr. 2022), pp. 3267–3274. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btac297. eprint: https://academic.oup.com/bioinformatics/article-pdf/38/12/3267/44045306/btac297.pdf. URL: https://doi.org/10.1093/bioinformatics/btac297.

[90] Alejandro Vaca Serrano, Guillem Garcia Subies, Helena Montoro Zamorano, et al. *RigoBERTa: A State-of-the-Art Language Model For Spanish*. 2022. arXiv: 2205.10233 [cs.CL].

[91] Asier Gutierrez-Fandino, Jordi Armengol-Estape, Marc Pamies, et al. "MarIA: Spanish Language Models". In: *Procesamiento del Lenguaje Natural* 68 (Mar. 2022), pp. 39–60. ISSN: 19897553. DOI: 10.26342/2022-68-3.

[92] Javier de la Rosa, Eduardo G. Ponferrada, Paulo Villegas, et al. *BERTIN: Efficient Pre-Training of a Spanish Language Model using Perplexity Sampling*. 2022. arXiv: 2207.06814 [cs.CL].

[93] Braja G Patra, Mohit M Sharma, Veer Vekaria, et al. "Extracting social determinants of health from electronic health records using natural language processing: a systematic review". In: *Journal of the American Medical Informatics Association* 28.12 (Oct. 2021), pp. 2716–2727. ISSN: 1527-974X. DOI: 10.1093/jamia/ocab170. eprint: https://academic.oup.com/jamia/article-pdf/28/12/2716/41325357/ocab170.pdf. URL: https://doi.org/10.1093/jamia/ocab170.

[94] Kevin Lybarger, Mari Ostendorf, and Meliha Yetisgen. "Annotating social determinants of health using active learning, and characterizing determinants using neural event extraction". In: *Journal of Biomedical Informatics* 113 (2021), p. 103631. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2020.103631. URL: https://www.sciencedirect.com/science/article/pii/S1532046420302598.

[95] Alistair EW Johnson, Tom J Pollard, Lu Shen, et al. "MIMIC-III, a freely accessible critical care database". In: *Scientific data* 3.1 (2016), pp. 1–9.

[96] *2022 n2c2 Shared Task and Workshop*. Accessed: October 2023. URL: https://n2c2.dbmi.hms.harvard.edu/2022-track-2.

[97] Russell Richie, Victor M Ruiz, Sifei Han, et al. "Extracting social determinants of health events with transformer-based multitask, multilabel named entity recognition". In: *Journal of the American Medical Informatics Association* (Apr. 2023). ocad046. ISSN: 1527-974X. DOI: 10.1093/jamia/ocad046. eprint: https://academic.oup.com/jamia/advance-article-pdf/doi/10.1093/jamia/ocad046/49725648/ocad046.pdf. URL: https://doi.org/10.1093/jamia/ocad046.

[98] Rachel Stemerman, Jaime Arguello, Jane Brice, et al. "Identification of social determinants of health using multi-label classification of electronic health record clinical notes". In: *JAMIA Open* 4.3 (Feb. 2021). ooaa069. ISSN: 2574-2531. DOI: 10.1093/jamiaopen/ooaa069. eprint: https://academic.oup.com/jamiaopen/article-pdf/4/3/ooaa069/40325411/ooaa069.pdf. URL: https://doi.org/10.1093/jamiaopen/ooaa069.

[99] Sifei Han, Robert F. Zhang, Lingyun Shi, et al. "Classifying social determinants of health from unstructured electronic health records using deep learning-based natural language processing". In: *Journal of Biomedical Informatics* 127 (2022), p. 103984. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2021.103984. URL: https://www.sciencedirect.com/science/article/pii/S1532046421003130.

[100] Zehao Yu, Xi Yang, Chong Dang, et al. *SODA: A Natural Language Processing Package to Extract Social Determinants of Health for Cancer Studies*. 2022. DOI: 10.48550/ARXIV.2212.03000. URL: https://arxiv.org/abs/2212.03000.

[101] Sripriya Rajamani, Elizabeth S Chen, Elizabeth Lindemann, et al. "Representation of occupational information across resources and validation of the occupational data for health model". In: *Journal of the American Medical Informatics Association* 25.2 (Apr. 2017), pp. 197–205. ISSN: 1527-974X. DOI: 10.1093/jamia/ocx035. eprint: https://academic.oup.com/jamia/article-pdf/25/2/197/34149953/ocx035.pdf. URL: https://doi.org/10.1093/jamia/ocx035.

[102] Lily A Cook, Jonathan Sachs, and Nicole G Weiskopf. "The quality of social determinants data in the electronic health record: a systematic review". In: *Journal of the American Medical Informatics Association* 29.1 (Oct. 2021), pp. 187–196. ISSN: 1527-974X. DOI: 10.1093/jamia/ocab199. eprint: https://academic.oup.com/jamia/article-pdf/29/1/187/41955552/ocab199.pdf. URL: https://doi.org/10.1093/jamia/ocab199.

[103] Laura A. McClure, Tulay Koru-Sengul, Monique N. Hernandez, et al. "Availability and accuracy of occupation in cancer registry data among Florida firefighters". In: *PLOS ONE* 14.4 (Apr. 2019), pp. 1–8. DOI: 10.1371/journal.pone.0215867. URL: https://doi.org/10.1371/journal.pone.0215867.

[104] Elizabeth A Lindemann, Elizabeth S Chen, Sripriya Rajamani, et al. "Assessing the Representation of Occupation Information in Free-Text Clinical Documents Across Multiple Sources." In: *Studies in health technology and informatics* 245 (2017), pp. 486–490. ISSN: 1879-8365.

[105] Annika M. Schoene, Ioannis Basinas, Martie van Tongeren, et al. "A Narrative Literature Review of Natural Language Processing Applied to the Occupational Exposome". In: *International Journal of Environmental Research and Public Health* 19.14 (2022). ISSN: 1660-4601. URL: https://www.mdpi.com/1660-4601/19/14/8544.

[106] *Exposome and Exposomics*. Aug. 2022. URL: https://www.cdc.gov/niosh/topics/exposome/default.html.

[107] Junhua Liu, Yung Chuen Ng, Zitong Gui, et al. "Title2Vec: a contextual job title embedding for occupational named entity recognition and other applications". In: *Journal of Big Data* 9 (1 Sept. 2022), p. 99. ISSN: 2196-1115. DOI: 10.1186/s40537-022-00649-5.

[108] Seda Kul and Ahmet Sayar. "Entity Name Recognition in Job Postings and Resumes". In: *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. 2021, pp. 1–6. DOI: 10.1109/HORA52670.2021.9461326.

[109] Miriam Mutambudzi, Claire Niedzwiedz, Ewan Beaton Macdonald, et al. "Occupation and risk of severe COVID-19: prospective cohort study of 120 075 UK Biobank participants". In: *Occupational and Environmental Medicine* 78.5 (2021), pp. 307–314. ISSN: 1351-0711. DOI: 10.1136/oemed-2020-106731. eprint: https://oem.bmj.com/content/78/5/307.full.pdf. URL: https://oem.bmj.com/content/78/5/307.

[110] Environmental Modelling Group. *EMG: COVID-19 risk by occupation and workplace, 11 February 2021*. 2021.

[111] Lukas Lange, Heike Adel, and Jannik Strötgen. *Boosting Transformers for Job Expression Extraction and Classification in a Low-Resource Setting*. 2021.

[112] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, et al. "Unsupervised Cross-lingual Representation Learning at Scale". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 8440–8451. DOI: 10.18653/v1/2020.acl-main.747. URL: https://aclanthology.org/2020.acl-main.747.

[113] Fazlourrahman Balouchzahi, Grigori Sidorov, and Hosahalli Lakshmaiah Shashirekha. "ADOP FERT-Automatic Detection of Occupations and Profession in Medical Texts using Flair and BERT." In: *IberLEF@ SEPLN*. 2021, pp. 747–757.

[114] José Canete, Gabriel Chaperon, Rodrigo Fuentes, et al. "Spanish pre-trained bert model and evaluation data". In: *Pml4dc at iclr* 2020 (2020), pp. 1–10.

[115] Alan Akbik, Tanja Bergmann, Duncan Blythe, et al. "FLAIR: An easy-to-use framework for state-of-the-art NLP". In: *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. 2019, pp. 54–59.

[116] M Dıaz-Galiano and L Alfonso Urena-López. "BERT Representations to Identify Professions and Employment Statuses in Health data". In: (2021).

[117] Montse Cuadros. "Vicomtech at MEDDOPROF: Automatic Information Extraction and Disambiguation in Clinical Text". In: (2021).

[118] Arantxa Otegi, Aitor Agirre, Jon Ander Campos, et al. "Conversational Question Answering in Low Resource Scenarios: A Dataset and Case Study for Basque". In: *Proceedings of The 12th Language Resources and Evaluation Conference*. 2020, pp. 436–442.

[119] Salvador Medina Herrera and Jorge Turmo Borras. "Everything transformers: Recognition, classification and normalisation of professions and family relations". In: *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2021): co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2021), XXXVII International Conference of the Spanish Society for Natural Language Processing: Málaga, Spain, September, 2021*. CEUR-WS. org. 2021, pp. 770–775.

[120] Kaushik Acharya. "Occupation Recognition and Normalization in Clinical Notes." In: *IberLEF@ SEPLN*. 2021, pp. 788–795.

[121] Jalaj Harkawat and Tejas Vaidhya. "Spanish Pre-Trained Language Models for HealthCare Industry." In: *IberLEF@ SEPLN*. 2021, pp. 796–802.

[122] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, et al. *LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention*. 2020. DOI: 10.48550/ARXIV.2010.01057. URL: https://arxiv.org/abs/2010.01057.

[123] Guillermo Lopez-Garcia, Jose M. Jerez, Nuria Ribelles, et al. "Transformers for Clinical Coding in Spanish". In: *IEEE Access* 9 (2021), pp. 72387–72397. ISSN: 21693536. DOI: 10.1109/ACCESS.2021.3080085.

[124] Alberto Blanco, Alicia Perez, and Arantza Casillas. "Exploiting ICD Hierarchy for Classification of EHRs in Spanish Through Multi-Task Transformers". In: *IEEE journal of biomedical and health informatics* 26 (3 Mar. 2022), pp. 1374–1383. ISSN: 2168-2208. DOI: 10.1109/JBHI.2021.3112130. URL: https://pubmed.ncbi.nlm.nih.gov/34520380/.

[125] Antonio Miranda-Escalada, Luis Gascó, Salvador Lima-López, et al. "Overview of DisTEMIST at BioASQ: Automatic detection and normalization of diseases from clinical texts: results, methods, evaluation and multilingual resources". In: (2022). DOI: 10.5281/zenodo.6408476. URL: https://doi.org/10.5281/zenodo.6408476.

[126] Vincenzo Moscato, Marco Postiglione, and Giancarlo Sperlí. *Biomedical Spanish Language Models for entity recognition and linking at BioASQ DisTEMIST*. 2022. URL: http://ceur-ws.org.

[127] Oswaldo Solarte Pabón, Orlando Montenegro, Maria Torrente, et al. "Negation and uncertainty detection in clinical texts written in Spanish: a deep learning-based approach". In: *PeerJ Computer Science* 8 (Mar. 2022), e913. ISSN: 2376-5992. DOI: 10.7717/peerj-cs.913.

[128] Ying Xiong, Yedan Shen, Yuanhang Huang, et al. "A Deep Learning-Based System for PharmaCoNER". In: Association for Computational Linguistics, 2019, pp. 33–37. DOI: 10.18653/v1/D19-5706.

[129] Aitor Gonzalez-Agirre, Montserrat Marimon, Ander Intxaurrondo, et al. *PharmaCoNER: Pharmacological Substances, Compounds and proteins Named Entity Recognition track*, pp. 1–10. URL: https://github.com/PlanTL-SANIDAD/SPACCC.

[130] Ying Xionga, Yuanhang Huanga, Qingcai Chena, et al. "A Joint Model for Medical Named Entity Recognition and Normalization". In: (2020). URL: https://ceur-ws.org/Vol-2664/cantemist_paper18.pdf.

[131] Antonio Miranda-Escalada, Eulàlia Farré, and Martin Krallinger. "Named Entity Recognition, Concept Normalization and Clinical Coding: Overview of the Cantemist Track for Cancer Text Mining in Spanish, Corpus, Guidelines, Methods and Results". In: (2020). DOI: 10.5281/zenodo.3773228. URL: https://doi.org/10.5281/zenodo.3773228.

[132] "Mention detection, normalization & classification of species, pathogens, humans and food in clinical documents: Overview of the LivingNER shared task and resources". In: (). ISSN: 1135-5948. DOI: 10.26342/2022-69-21. URL: http://hdl.handle.net/10045/127432.

[133] Elena Zotova, Aitor García-Pablos, Naiara Perez, et al. *Vicomtech at LivingNER2022*. 2022. URL: http://ceur-ws.org.

[134] Daniel Bravo-Candel, Jésica López-Hernández, José Antonio García-Díaz, et al. "Automatic correction of real-word errors in Spanish clinical texts". In: *Sensors* 21 (9 May 2021). ISSN: 14248220. DOI: 10.3390/s21092893.

[135] Xabier Soto, Olatz Perez-De-Viñaspre, Maite Oronoz, et al. *Leveraging SNOMED CT terms and relations for machine translation of clinical texts from Basque to Spanish*. URL: https://www.modela.eus/eu/itzultzailea.

[136] Oswaldo Solarte-Pabón, Orlando Montenegro, Alvaro García, et al. *Highlights A Deep Learning Approach to Extract Lung Cancer Information from Spanish clinical texts*. 2022. DOI: 10.2139/ssrn.4049602. URL: https://ssrn.com/abstract=4049602.

[137] Eduardo Godoy, Steren Chabert, Marvin Querales, et al. "A named entity recognition framework using transformers to identify relevant clinical findings from mammographic radiological reports". In: ed. by Jorge Brieva, Pamela Guevara, Natasha Lepore, et al. Vol. 12567. SPIE, 2023, p. 125670X. DOI: 10.1117/12.2670228. URL: https://doi.org/10.1117/12.2670228.

[138] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, et al. "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*. Avignon, France: Association for Computational Linguistics, Apr. 2012, pp. 102–107. URL: https://aclanthology.org/E12-2021.

[139] Montserrat Marimon, Aitor Gonzalez-Agirre, Ander Intxaurrondo, et al. *MEDDOCAN corpus: gold standard annotations for Medical Document Anonymization on Spanish clinical case reports*. Version 1.0. Funded by the Plan de Impulso de las Tecnologías del Lenguaje (Plan TL). Zenodo, Nov. 2020. DOI: 10.5281/zenodo.4279323. URL: https://doi.org/10.5281/zenodo.4279323.

[140] Salvador Lima Lopez, Naiara Perez, Montse Cuadros, et al. "NUBes: A Corpus of Negation and Uncertainty in Spanish Clinical Texts". English. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 5772–5781. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.708.

[141] Montserrat Marimon, Jorge Vivaldi, and Núria Bel. "Annotation of negation in the IULA Spanish Clinical Record Corpus". In: *Proceedings of the Workshop Computational Semantics Beyond Events and Roles*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 43–52. DOI: 10.18653/v1/W17-1807. URL: https://aclanthology.org/W17-1807.

[142] Mercedes Aguado and Núria Bel Rafecas. *A Corpus of Spanish clinical records annotated for abbreviation identification*. 2022. URL: http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6409.

[143] Ander Intxaurrondo, Juan Carlos de la Torre, Montserrat Marimon, et al. "Resources, guidelines and annotations for the recognition, definition resolution and concept normalization of Spanish clinical abbreviations: the BARR2 corpus". In.

[144] Antonio Miranda-Escalada, Eulàlia Farré, and Martin Krallinger. *Cantemist corpus: gold standard of oncology clinical cases annotated with CIE-O 3 terminology*. Version 1.6. Funded by the Plan de Impulso de las Tecnologías del Lenguaje (Plan TL). Zenodo, Apr. 2020. DOI: 10.5281/zenodo.3978041. URL: https://doi.org/10.5281/zenodo.3978041.

[145] Antonio Miranda-Escalada, Aitor Gonzalez-Agirre, and Martin Krallinger. *CodiEsp corpus: gold standard Spanish clinical cases coded in ICD10 (CIE10) - eHealth CLEF2020*. Version 1.4. Funded by the Plan de Impulso de las Tecnologías del Lenguaje (Plan TL). Zenodo, May 2020. DOI: 10.5281/zenodo.3837305. URL: https://doi.org/10.5281/zenodo.3837305.

[146] Antonio Miranda-Escalada, Eulàlia Farré-Maduell, Gloria González Gacio, et al. *LivingNER corpus: Named entity recognition, normalization & classification of species, pathogens and food*. Version 6.3.1. Funded by the Plan de Impulso de las Tecnologías del Lenguaje (Plan TL). June 2022. DOI: 10.5281/zenodo.6768606. URL: https://doi.org/10.5281/zenodo.6768606.

[147] Aitor Gonzalez-Agirre, Montserrat Marimon, Ander Intxaurrondo, et al. "PharmaCoNER: Pharmacological Substances, Compounds and proteins Named Entity Recognition track". In: *Proceedings of the 5th Workshop on BioNLP Open Shared Tasks*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 1–10. DOI: 10.18653/v1/D19-5701. URL: https://aclanthology.org/D19-5701.

[148] Antonio Miranda-Escalada, Eulàlia Farré, Luis Gasco, et al. *DisTEMIST corpus: detection and normalization of disease mentions in spanish clinical cases*. Version 5.1. Funded by the Plan de Impulso de las Tecnologías del Lenguaje (Plan TL). Zenodo, June 2022. DOI: 10.5281/zenodo.6671292. URL: https://doi.org/10.5281/zenodo.6671292.

[149] Mariia Chizhikova, Pilar López-Úbeda, Jaime Collado-Montañez, et al. "CARES: A Corpus for classification of Spanish Radiological reports". In: *Computers in Biology and Medicine* 154 (2023), p. 106581. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2023.106581. URL: https://www.sciencedirect.com/science/article/pii/S001048252300046X.

[150] Pablo Báez, Fabián Villena, Matías Rojas, et al. *The Chilean Waiting List Corpus*. Zenodo, Nov. 2020. DOI: 10.5281/zenodo.7555181. URL: https://doi.org/10.5281/zenodo.7555181.

[151] Maite Oronoz, Koldo Gojenola, Alicia Pérez, et al. "On the creation of a clinical gold standard corpus in Spanish: Mining adverse drug reactions". In: *Journal of Biomedical Informatics* 56 (2015), pp. 318–332. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2015.06.016. URL: https://www.sciencedirect.com/science/article/pii/S1532046415001264.

[152] Noa Cruz, Roser Morante, Manuel J. Maña López, et al. "Annotating Negation in Spanish Clinical Texts". In: *Proceedings of the Workshop Computational Semantics Beyond Events and Roles*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 53–58. DOI: 10.18653/v1/W17-1808. URL: https://aclanthology.org/W17-1808.

[153] Casimiro Pio Carrino, Jordi Armengol-Estapé, Ona de Gibert Bonet, et al. *Spanish Biomedical Crawled Corpus: A Large, Diverse Dataset for Spanish Biomedical Language Models*. 2021. DOI: 10.48550/ARXIV.2109.07765. URL: https://arxiv.org/abs/2109.07765.

[154] Isabel Segura-Bedmar, Ricardo Revert, and Paloma Martínez. "Detecting drugs and adverse events from Spanish social media streams". In: *Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis (Louhi)*. Gothenburg, Sweden: Association for Computational Linguistics, Apr. 2014, pp. 106–115. DOI: 10.3115/v1/W14-1117. URL: https://aclanthology.org/W14-1117.

[155] Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, et al. "PhysioBank, PhysioToolkit, and PhysioNet". In: *Circulation* 101.23 (2000), e215–e220. DOI: 10.1161/01.CIR.101.23.e215. eprint: https://www.ahajournals.org/doi/pdf/10.1161/01.CIR.101.23.e215. URL: https://www.ahajournals.org/doi/abs/10.1161/01.CIR.101.23.e215.

[156] Eulàlia Farré-Maduell, Salvador Lima-López, Antonio Miranda-Escalada, et al. *MEDDOPROF guidelines*. Apr. 2021. DOI: 10.5281/zenodo.4720833. URL: https://doi.org/10.5281/zenodo.4720833.

[157] Mariana Neves and Jurica Ševa. "An extensive review of tools for manual annotation of documents". In: *Briefings in Bioinformatics* 22.1 (Dec. 2019), pp. 146–163. ISSN: 1477-4054. DOI: 10.1093/bib/bbz130. eprint: https://academic.oup.com/bib/article-pdf/22/1/146/35934687/supplementary_data_for_the_survey_annotation_tool_bbz130.pdf. URL: https://doi.org/10.1093/bib/bbz130.

[158] Davy Weissenbacher, Karen O'Connor, Aiko T. Hiraki, et al. "An empirical evaluation of electronic annotation tools for Twitter data". In: *Genomics Inform* 18.2 (2020), e24–. DOI: 10.5808/GI.2020.18.2.e24. eprint: http://genominfo.org/journal/view.php?number=612. URL: http://genominfo.org/journal/view.php?number=612.

[159] Wes McKinney. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. "O'Reilly Media, Inc.", 2012.

[160] Thomas Wolf, Lysandre Debut, Victor Sanh, et al. "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6. URL: https://aclanthology.org/2020.emnlp-demos.6.

[161] F. Pedregosa, G. Varoquaux, A. Gramfort, et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[162] Adam Paszke, Sam Gross, Francisco Massa, et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

[163] Martín Abadi, Ashish Agarwal, Paul Barham, et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[164] Francois Chollet et al. *Keras*. 2015. URL: https://github.com/fchollet/keras.

[165] Hiroki Nakayama. *seqeval: A Python framework for sequence labeling evaluation*. Software available from https://github.com/chakki-works/seqeval. 2018. URL: https://github.com/chakki-works/seqeval.

[166] Ekaba Bisong. *Google Colaboratory*. Apress, 2019, pp. 59–64. DOI: 10.1007/978-1-4842-4470-8_7.

[167] Franck Dernoncourt, Ji Young Lee, and Peter Szolovits. "NeuroNER: an easy-to-use program for named-entity recognition based on neural networks". In: *Conference on Empirical Methods on Natural Language Processing (EMNLP)* (2017).

[168] Martin Popel and Ondřej Bojar. "Training Tips for the Transformer Model". In: *The Prague Bulletin of Mathematical Linguistics* 110 (1 May 2018), pp. 43–70. DOI: 10.2478/pralin-2018-0002.

[169] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2017. DOI: 10.48550/ARXIV.1711.05101. URL: https://arxiv.org/abs/1711.05101.

[170] Dell Zhang, Jun Wang, and Xiaoxue Zhao. "Estimating the Uncertainty of Average F1 Scores". In: *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*. ICTIR '15. Northampton, Massachusetts, USA: Association for Computing Machinery, 2015, pp. 317–320. ISBN: 9781450338332. DOI: 10.1145/2808194.2809488. URL: https://doi.org/10.1145/2808194.2809488.

[171] CVC. Centro Virtual Cervantes. *CVC. Anuario 2022. informe 2022. El Español en cifras.* Oct. 2022. URL: https://cvc.cervantes.es/lengua/anuario/anuario_22/informes_ic/p01.htm.

[172] Jianing Wang, Nuo Chen, Qiushi Sun, et al. *HugNLP: A Unified and Comprehensive Library for Natural Language Processing.* 2023. arXiv: 2302.14286 [cs.CL].

[173] Shufang Xie, Huishuai Zhang, Junliang Guo, et al. *ResiDual: Transformer with Dual Residual Connections.* 2023. arXiv: 2304.14802 [cs.CL].

[174] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing. Vienna, Austria, 2022. URL: https://www.R-project.org/.

[175] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, et al. *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations.* 2020. arXiv: 1909.11942 [cs.CL].