1 2 9 0

UNIVERSIDADE Đ
COIMBRA

Fábio André da Costa Lopes

Contributions to Clinical Information Extraction in Portuguese: Corpora, Named Entity Recognition, Word Embeddings

Thesis submitted to the Faculty of Science and Technology of the University of Coimbra for the degree of Master in Biomedical Engineering with specialization in Clinical Informatics and Bioinformatics, supervised by Prof. Dr. Hugo Gonçalo Oliveira and Prof. Dr. César Teixeira.

July 2019

1 2 9 0

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE Ð
COIMBRA

Fábio André da Costa Lopes

Contributions to Clinical Information Extraction in Portuguese: Corpora, Named Entity Recognition, Word Embeddings

Thesis submitted to the
University of Coimbra for the degree of
Master in Biomedical Engineering

Supervisors:
Prof. Dr. Hugo Gonçalo Oliveira (CISUC)
Prof. Dr. César Teixeira (CISUC)

Coimbra, 2019

This work was developped in collaboration with:

CISUC - Center for Informatics and Systems of the University of Coimbra

CHUC - Coimbra Hospital and University Centre

Agradecimentos

Em primeiro lugar, gostaria de agradecer aos meus orientadores Professor Doutor Hugo Gonçalo Oliveira e Professor Doutor César Teixeira pela disponibilidade que tiveram desde o início deste projeto, por todo o apoio e motivação dados durante a sua realização e por todas as aprendizagens que me proporciaram ao longo deste ano letivo.

Quero agradecer ao Dr. Francisco Sales e à Dra. Catarina Cruto pela sua disponibilidade aquando da recolha do conjunto de dados do Serviço de Neurologia do Centro Hospitalar Universidade de Coimbra.

Agradeço a todos os meus colegas de laboratório por toda a ajuda dada durante este ano letivo, tanto no esclarecimento de dúvidas como também na motivação na realização do projeto. Também lhes agradeço por todas as conversas partilhadas e por me terem aturado durante este ano letivo.

Um grande obrigado a todos os amigos por todas as gargalhadas, momentos vividos e por todas as aprendizagens que me proporcionaram. Sem dúvida que todos estes anos não teriam sido os mesmos sem a vossa presença. Um agradecimento especial aos meus dois colegas de casa que foram como uma segunda família para mim, ao meu colega açoriano por todas as conversas partilhadas e aos meus dois companheiros de ginásio que sempre me acompanharam e motivaram não só na vida académica como também no desporto.

Um grande obrigado à Daniela por todo o apoio e momentos vividos durante estes anos tanto nos momentos bons como naqueles mais difíceis. É difícil exprimir em palavras o valor que tu tens para mim!

Por fim, deixo o meu maior agradecimento aos meus pais e irmã por todo o apoio que me deram em todos os momentos, bons e maus, ao longo do meu percurso. Obrigado por todo o esforço que fizeram para que nunca me faltasse nada durante estes vinte e dois anos. Certamente esse esforço não foi em vão!

“Those who can imagine anything, can create the impossible.”

ALAN TURING

Resumo

O grande aumento do uso de Registos Médicos Eletrónicos, por todo o mundo, levou a um crescimento exponencial da informação clínica. Só no sistema de saúde português, o uso destes nos hospitais aumentou de 42% para 83% entre 2004 e 2014. Contudo, tal informação é escrita em formatos não estruturados o que torna difícil o seu processamento. Apesar da solução para extrair dados seria fazê-lo manualmente, isto não só requer treinar técnicos de saúde, para efetuar tal tarefa, como também é uma solução intensiva que exige muito tempo. É nisto que a inteligência artificial pode ser útil permitindo construir modelos que permitem extrair informação automaticamente. Uma importante parte deste processo envolve o reconhecimento de entidades significativas no texto e, portanto, o desenvolvimento de modelos de reconhecimento de entidades mencionadas.

Para tal, o trabalho descrito nesta tese compreende seis tarefas principais: anotação de entidades mencionadas em texto clínico português; criação de um modelo de *Word Embeddings (WEs)* treinado com textos clínicos portugueses e comparar a sua performance com um modelo de *WEs* treinado com um grande conjunto de textos gerais que não são focados no domínio clínico; estudar as melhores características para reconhecimento de entidades mencionadas em texto clínico; analisar a performance de um modelo treinado em textos de casos clínicos recolhidos de uma revista médica quando testado em um conjunto de teste independente do anterior de textos recolhidos do serviço de Neurologia do Centro Hospitalar da Universidade de Coimbra.

Os modelos de reconhecimento de entidades mencionadas obtiveram medidas F1 de aproximadamente 83% e 75% para avaliação relaxada e e rigorosa, respetivamente, nos textos extraídos da revista médica. Para os textos de teste, as medidas F1 para a avaliação relaxada e rigorosa foram 71.21% e 62.71%, respetivamente. Concluímos também que os modelos de aprendizagem profunda obtém melhores resultados que os modelos de aprendizagem superficial e que, os modelos de *WE* treinados com texto clínico obtêm melhores resultados que os que são treinados com texto geral, mesmo que o último tenha sido treinados com muito mais textos que o primeiro. Além disso, os nossos reusltados mostram que é possível extrair informação de textos clínicos do Hospital com modelos treinados com casos clínicos extraídos de revistas clínicas públicas. Contudo, tais resultados ainda requerem um técnico de saúde para analisar se a informação é extraída corretamente.

Palavras-Chave: Processamento de Linguagem Natural, Aprendizagem Máquina, Reconhecimento de Entidades Mencionadas, Texto Clínico Português

Abstract

The great increase of using Electronic Medical Records (EMRs) in all world lead to an exponential growth of clinical information. Considering Portugal healthcare system, the use of EMRs in the hospitals rose from 42% to 83% from 2004 to 2014. However, such information is written in an unstructured way which is difficult to process. Although a solution for extracting such data would be doing it manually, it does not only require training healthcare technicians for doing so, but it is also a time consuming and intensive task. This is where Artificial Intelligence (AI) can be useful by making models that are able to perform Information Extraction (IE) automatically. An important part of this process involves recognizing meaningful entities in text, and thus the development of Named Entity Recognition (NER) models.

Towards the previous, the work described in this thesis comprised six main tasks: annotation of Named Entities (NEs) in Portuguese clinical texts; creation of a WE model trained with Portuguese clinical texts and comparison of its performance with a WE model trained in a large set of general-language texts; study of the best features for clinical NER; comparison between shallow machine learning classifiers with deep learning models; analyse the performance of a model trained on clinical case texts extracted from a medical journal in a independent test set of texts from the Coimbra Hospital and Universitary Centre (CHUC) Neurology Service.

Models for NER achieved F1-Scores of nearly 83% and 75%, respectively for relaxed and strict evaluation, on texts extracted from the medical journal. For texts collected from the Hospital, the same F1-Scores were 71.21% and 62.71%. We also conclude that deep learning models outperform the shallow models and that in-domain WEs get better results that out-of-domain ones, even when the latter were trained with much more texts than the former. Furthermore, our results show that it is possible to extract information from Hospital clinical texts with models trained with clinical cases extracted from journals, and thus openly available. However, such results still require a healthcare technician to check if the information is well extracted.

Keywords: Natural Language Processing, Machine Learning, Named Entity Recognition, Portuguese Clinical Text

Contents

List of Figures

List of Tables

List of Abbreviations

AI Artificial Intelligence. xiii, xvii, 2, 7, 14, 15

AR Agreement Ratio. 21, 35, 37

BiLSTM Bidirectional Long Short-Term Memory. xvii, xviii, xxi, 24, 25, 26, 27, 28, 45, 46, 47, 48, 49, 52, 53, 54, 59, 60, 62, 63, 67, 69, 70

CBOW Continuous Bag-of-Words. 13

CHUC Coimbra Hospital and Universitary Centre. xiii, xvi, 2, 3, 4, 31, 63

CNN Convolutional Neural Network. 10, 15, 24, 27, 28

CRF Conditional Random Fields. xvii, xviii, xxi, 3, 7, 10, 11, 12, 23, 24, 25, 26, 27, 28, 31, 40, 42, 45, 46, 47, 48, 49, 50, 52, 53, 54, 59, 60, 62, 63, 65, 66, 67, 69, 70

CV Cross Validation. xvii, xviii, 18, 49, 50, 51, 52, 54, 59

EMR Electronic Medical Record. xiii, 1, 24, 26, 27

FFN Feed-Forward Network. 15, 24, 27, 28

GRU Gated Recurrent Unit. 15, 24, 27, 28

HMM Hidden Markov Models. 10, 11, 23

IE Information Extraction. xiii, 2, 3, 7, 8, 9, 15, 21, 70

IOB Inside-Outside-Beginning. xvii, 9, 34, 42, 52

L-BFGS Limited-Memory Broyden–Fletcher–Goldfarb–Shanno. 43, 50

LSTM Long Short-Term Memory. xvii, xviii, 3, 15, 16, 17, 24, 27, 28, 47, 52, 53, 54, 67

ME Maximum Entropy. 10, 11, 12

MEMM Maximum Entropy Markov Models. xvii, 11, 12, 23

NB Naïve Bayes. 10, 11

NE Named Entity. xiii, xvii, xxi, 3, 9, 10, 21, 22, 23, 32, 33, 34, 35, 37, 38, 37, 43, 44, 45, 47, 50, 59, 63, 65, 66, 70

NER Named Entity Recognition. xiii, xxi, 2, 3, 4, 7, 8, 9, 10, 21, 22, 23, 24, 25, 26, 27, 28, 29, 31, 35, 39, 45, 59, 63, 69, 70
NLP Natural Language Processing. 2, 3, 4, 7, 8, 15, 24, 32, 35, 69, 70
NPMI Normalized Pointwise Mutual Information. 23, 42

OECD Organisation for Economic Co-operation and Development. 1

POS Part-of-Speech. 8, 23, 26, 32, 34, 42, 46, 52

RE Relation Extraction. 2, 8
RNN Recurrent Neural Network. xvii, 3, 7, 10, 15, 16, 24, 25, 27, 28

SVM Support Vector Machines. 10, 23, 24

WE Word Embedding. xi, xiii, xviii, xxi, 4, 13, 14, 21, 23, 24, 25, 26, 31, 39, 40, 42, 46, 48, 49, 52, 53, 54, 55, 57, 58, 65, 69, 70

1

Introduction

This chapter explains a brief motivation of this thesis which includes some statistics about the theme of the project in section 1.1 and the context of the thesis in section 1.2. Then we present the goals of this project and how we are going to fulfil them in section 1.3. Furthermore, we present the contributions of this project for the scientific community (section 1.4) and finally we introduce the structure of this document in the last section of this chapter.

1.1 Motivation

The exponential growth of the computational resources allowed the increase of data production and storage on different areas such as economy, sports or industry. According to the Organisation for Economic Co-operation and Development (OECD) one of the most important is the healthcare area [1], which, besides its general relation to well-being, is also economically-relevant.

In the last years, we have seen a great increase of using Electronic Medical Records (EMRs) across all the world. In 2017, in the United States of America, 85.9% office-based physicians use a EMR system [1]. In 2016, a survey was made on 15 European Union countries that belong to OECD concluding that about 80% on average use EMR on primary care practices [2]. In Portugal, from 2004 to 2014, the number of hospitals using EMRs rose from 42% to 83%[2], which means that there is much electronic clinical information available.

[1] https://www.cdc.gov/nchs/fastats/electronic-medical-records.htm

[2] https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_destaques&DESTAQUESdest_boui=222174618&DESTAQUESmodo=2

1.2 Context

EMR is a collection of clinical information about a patient. It includes the clinical status e.g. the height, the weight, the blood pressure and allergies. It also contains reports written by physicians, nurses and healthcare technicians, e.g., x-ray reports, nurse's notes, physician's notes and diagnostic image reports.

Such reports contain information about clinical events of the patients, e.g., their results of diagnostic tests, their clinical evolution or the most frequent diseases during a time span [3]. It is thus valuable information, e.g., for making association rules between diseases, therapies and diagnostic exams, which could be very important, not only for the hospitals but also for the pharmaceutical industry. However, such information is often written in natural language and thus not structured for immediate consumption. This is where Artificial Intelligence (AI) can be useful.

We have been seeing a growth of AI on the clinical area, e.g., in the support for diagnosis [4, 5] and in surgical interventions [6]. However, for clinical area improvement, it is also important to leverage on written data, and where Natural Language Processing (NLP) techniques should be exploited towards a better interpretation of patient clinical reports.

NLP is a branch of AI that deals with human language manipulation and understanding by machines, e.g., extracting information from texts as it is written in the last paragraph and relate it in order to summarise the clinical reports using natural language generation [7], classifying texts with proper classes such as whether a comment is positive or negative [8] or word sense disambiguation, which is responsible to check what is the desired meaning of a word in a certain sentence [9]. Furthermore, inside this branch there are two relevant sub-tasks when it comes to performing Information Extraction (IE): the Named Entity Recognition (NER) and Relation Extraction (RE). While NER is responsible for recognising concepts that are relevant for interpreting the meaning of text and classifying them according to a set of categories, RE identifies meaningful relations between those concepts.

IE tools enable the extraction of relevant information in a structured way, which thus become more accessible for humans or further analysis by computational tools, which saves time for healthcare workers, who will not have to collect the information manually. NLP also provides a way to discover hidden patterns which could go unnoticed for humans.

1.3 Goals

This study was originally triggered by our access to a collection of clinical documents from the Neurology service of the Coimbra Hospital and Universitary Centre (CHUC), in Coimbra, Portugal, and thus written in Portuguese. While willing to extract information from those texts, we quickly noticed that, although clinical text mining is not a new area, and there is much work in English written tests, Portuguese NLP studies are still scarce [10, 11], which motivated us to contribute to the area at different levels.

Given that languages are different, i.e., have different grammars, vocabulary, follow different writing styles, we aim to study whether it is possible to adopt state-of-the-art methods in NER from Portuguese clinical data, for its future application in CHUC.

To the best of our knowledge as there were no Portuguese datasets available it was mandatory to collect public clinical texts which could be used for the creation of the NER models. After that, it was necessary to convert each of them to a structured form and finally as they were collected in raw it was required to manual labelling them with the Named Entity (NE) classes which we want to extract from the texts.

Our proposal for creating the NER model consists of two different algorithms, the Conditional Random Fields (CRF) and Recurrent Neural Network (RNN) based on Long Short-Term Memory (LSTM) layers. As the first uses interpretable features it allows us to study what are the best features as well as the best transition rules in the IE task. The second algorithm was used because it is the actual state-of-art algorithm for the IE task. However, as it is a deep learning algorithm it is seen as a black box which means that we are not able to interpret what is the model doing when it does the classification.

Furthermore, we are going to compare a out-of-domain distributional semantic model with a model trained with only clinical texts collected from a Portuguese Neurology journal called Sinapse.

1.4 Contributions

This thesis contributes with valuable resources in NLP field such as:

- Availability of 281 clinical texts manually labelled with 13 different NEs categories;
- Availability of a Word Embedding (WE) model pre-trained with 3377 clinical texts;
- The best features for training NER models in clinical field;
- Comparison between a WE model trained with clinical texts and a WE model downloaded from FastText repository with was trained with general texts;
- Comparison of the performance of different algorithms in the NER task;
- Verification of the performance of a model trained with texts of a journal in extracting information from CHUC texts.

This project also contributed with two papers published in two international conferences:

- **Fábio Lopes**, César Teixeira and Hugo Gonçalo Oliveira, "Named Entity Recognition in Portuguese Neurology Text using CRF", EPIA 2019 - 19th EPIA Conference on Artificial Intelligence (Accepted as full paper on 7th June to present on Vila Real, Portugal on 3rd-6th September, 2019)
- **Fábio Lopes**, César Teixeira and Hugo Gonçalo Oliveira, "Contributions to Clinical Named Entity Recognition in Portuguese", BioNLP 2019 - 18th ACL Workshop on Biomedical Natural Language Processing (Accepted as full paper on 31st May to present on Florence, Italy on 1st August, 2019)

1.5 Structure

This document contains 5 chapters beyond the introduction. **Chapter 2** presents a brief explanation of each concept used in this thesis as what is NLP or how the different machine learning algorithms work. Then, in **Chapter 3** we discuss about the old state of the art models and the current ones comparing the advantages and disadvantages of each one as well as their performances. **Chapter 4** describes the dataset including its annotation and the statistics about the labels. In addition, this chapter introduces the feature processing methods and the parameters of each model. Afterwards, **Chapter 5** contains the grid search results for each model as well as the validation results. Then, the chapter discusses about which model is the best using not only average results comparison but also statistical test results. Finally, there is a detailed discussion about the results of the models on an independent test set. Finally, **Chapter 6** concludes with the achieved goals, limitations of the work and also the future work.

2

Background Concepts

In this chapter, the main concepts required to understand this work are introduced. Section 2.1 presents the concept of Natural Language Processing (NLP) as well as tasks where it is successfully used. One of these tasks is Information Extraction (IE) which is responsible for retrieving information from unstructured sources as text. We also present how a text is processed from its raw form to a structured form. Then, section 2.2 gives a brief explanation of Named Entity Recognition (NER) using not only concepts but also some examples. It is also discussed what are the models used to perform NER. Section 2.4 provides a simple description of vector semantics as well as the algorithms used in this task. Sections 2.3 and 2.5 present the algorithms used in this work, namely the Conditional Random Fields (CRF) and Recurrent Neural Networks (RNNs). Finally, this chapter closes with a simple description of what metrics were used for evaluate the results.

2.1 Natural Language Processing

NLP [12] is a field of Artificial Intelligence (AI) that aims to enable machines to understand the human language and to use it to communicate with the humans. It allows machines to get data from texts and human speech. It is not an easy task due to the ambiguity [13] that exists in natural language. For instance, there are many words and expressions that have different meaning for different situations e.g. "see eye to eye" or "when pigs fly" which mean when people agree with each other and something that is never going to happen, respectively. Another problem related to NLP is the linguistic variation, i.e., people have many different ways to say a certain concept or expression, e.g., the word "brain" could be described as the "center of the nervous system" or the "organ of intelligence". Although for humans these phrases are easy to understand because of their knowledge, for computers they are not. NLP can be seen in real life in tasks as information retrieval [14], IE [15], machine translation [16], text simplification [17], sentiment analysis [18], text summarization [19], spam filter [20], natural language generation [21], speech recognition [22] or in chatbots for question answering [23].

IE allows to get data from unstructured sources such as images, audio or video. However, we will focus on getting data from texts as it is the main goal of our work. Getting the data from unstructured sources to structured repositories enables an easier processing by machines. IE could be used in any type of text e.g. news, biomedical or business [24]. For getting the information from raw texts they first have to be preprocessed. The preprocessing steps include: (1) tokenization, (2) Part-of-Speech (POS) tagging and also (3) lemmatisation, although the latter is not always necessary. Tokenization is responsible for word segmentation, in other words, it splits the text into individual words or punctuation [25]. POS tagging is the process of labelling each word with its part-of-speech function, e.g. noun, verb, abverb or adjective [26]. Lemmatisation is the process where each token is reduced to its dictionary form. This step allows the NLP models to group words with the same root e.g. the words walked, walk, walks and walking are all reduced to the word walk [25]. The three steps are illustrated in figure 2.1.

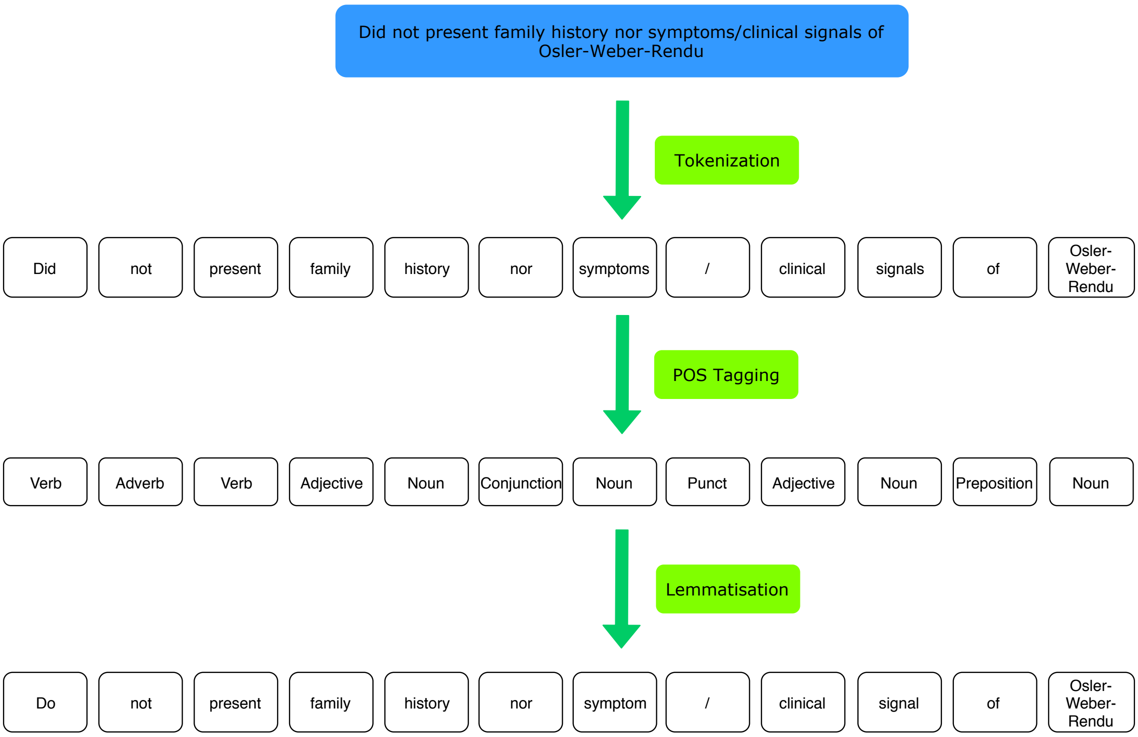


Figure 2.1: Preprocessing pipeline on the sentence: “Did not present family history nor symptoms/clinical signals of Osler-Weber-Rendu”.

After preprocessing, NER and Relation Extraction (RE) are done for getting all the important information from the unstructured sources.

2.2 Named Entity Recognition

NER is a subtask of IE from Text that is used for detecting and classifying Named Entities (NEs) in the text as it is shown in figure 2.2.

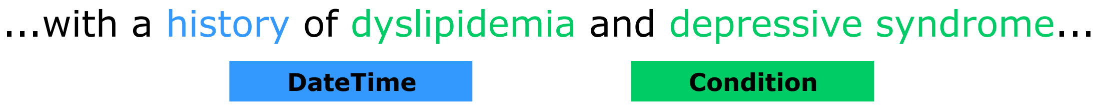
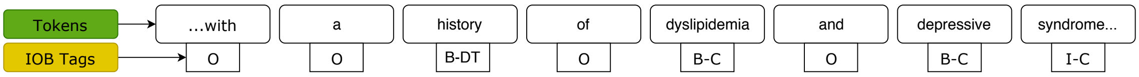

...with a history of dyslipidemia and depressive syndrome...
DateTime
Condition

Figure 2.2: NE example on the sentence: "...with a history of dislipidemia and depressive syndrome...". "history" belongs to DateTime NE and "dyslipidemia" and "depressive syndrome" belong to Condition NE.

A NE is the instantiation of a specific type of concept that is relevant for understanding the meaning of the text, e.g. in the general case: names, organizations, locations and time expressions; and in the clinical field: names of diagnostic tests, diseases and therapeutics. This task has some difficulties as classifying equal words in different contexts, e.g. the word "apple" could be a fruit or the name of a company. For classifying this type of words it is important to use not only features from the current word but also from surrounding words [27]. To be easier for the NER model to know when a word is inside an NE, during the process of classifying each word a sequential tagging system should be used. A popular system, also used in this work, is the Inside-Outside-Beginning (IOB) tagging. It has three different types of tags: the beginning tag (B), the inside tag (I) and the outside tag (O). The beginning tag means that the token is beginning a certain NE. The inside tag means that the token is inside a NE and therefore, it is mandatory to have a beginning tag or other inside tag from the same NE before this tag. Finally, the outside tag means that the token does not belong to any NE [24, 28]. This way of tagging the tokens is useful for the classifier to create specific transition rules, e.g. the inside tag never appears before a beginning tag of the same NE. This type of tagging is shown in figure 2.3.



Tokens	...with	a	history	of	dyslipidemia	and	depressive	syndrome...
IOB Tags	O	O	B-DT	O	B-C	O	B-C	I-C

Figure 2.3: IOB tagging example on the sentence: "...with a history of dyslipidemia and depressive syndrome...". Reference: O: Out; B-DT: Begin-DateTime; B-C: Begin-Condition; I-C: In-Condition.

NER is a task that can be performed using the following different types of models:

- Rule-based, that try to directly match the words using manually or automatically pre-built rules/patterns [29];
- Models that try to find words in the text that appear in a well-known nomenclature dictionary, often called gazetteers [30, 31];
- Shallow Machine-learning approaches that try to find the correct NE class for each token based on manually-created features. CRF and Structured Support Vector Machines (SVM) are examples of shallow machine learning algorithms used to build the NER model [32, 33, 34];
- Deep Learning models that classifies each token based on its embedding vector. RNN and Convolutional Neural Network (CNN) are examples of algorithms used to build these models [35, 36];
- Hybrid approaches that ensemble two or more of the above ways of doing NER [37, 38].

2.3 Conditional Random Fields

CRF is a probabilistic model that is based on ideas taken from other probabilistic models such as Naïve Bayes (NB), Hidden Markov Models (HMM) and Maximum Entropy (ME). However, while NB and HMM are generative models that try the maximize the joint likelihood $p(y,\vec{x})$, ME is a discriminative model, i.e., it learns the conditional probability distribution $p(y|\vec{x})$. NB classifies single class instances based on several feature values. It considers that the features are conditionally independent from each other. The model is based on the Bayes' theorem (equation 2.1):

$$p(y|\vec{x}) = \frac{p(y)p(\vec{x}|y)}{p(\vec{x})} \tag{2.1}$$

where

$P(y)$: Class Prior Probability given by the ratio of the number of samples of y and the total number of samples
$P(\vec{x}|y)$: Likelihood
$P(\vec{x})$: Evidence or Predictor Prior Probability

HMM [39] is a simple approach based on NB to classify sequences. However, instead of using several features for classifying each sequence position, HMM only uses one feature, as shown in equation 2.2. It is a hardly used algorithm for creating NER models since each instance of the sequence only depends on the labels from the previous observation and the current observation and on the current observation.

$$p(\vec{y},\vec{x}) = \prod_{i=0}^{n} p(y_i|y_{i-1})p(x_i|y_i) \tag{2.2}$$

ME is a conditional probability model based on the Principle of the Maximum Entropy [40]. Its objective during training is to find the largest possible conditional entropy while being consistent with the training data. Basically, ME is to Maximum Entropy Markov Models (MEMM) and CRF what NB is to HMM [41]. The model is based on equation 2.3. Feature functions f($\vec{x}$,y) behave like rules and their weights (λ) represent whether the function promotes the predicted label or not. An example is shown on equation 2.4.

$$P(y|\vec{x}) = \frac{\exp\sum_i \lambda_i f_i(\vec{x},y)}{\sum_{y'\in Y}\exp\sum_i \lambda_i f_i(\vec{x},y')} \tag{2.3}$$

$$f_i(\vec{x},y) = \begin{cases} 1 & \text{if word=“diabetes” and next word=“mellitus” and y=“Condition”} \\ 0 & \text{else} \end{cases} \tag{2.4}$$

MEMM [42] is a directed graphical sequential classifier that is based on ME which makes it discriminative as well. As it is based on ME it allows to do sequence labelling using several features which improve the classification. It is described by the equation 2.5.

$$p(\vec{y}|\vec{x}) = \prod_i \frac{\exp\sum_j \lambda_j f_j(y_i,y_{i-1},\vec{x},i)}{\sum_{y'\in Y}\exp\sum_j \lambda_j f_j(y',y_{i-1},\vec{x},i)} \tag{2.5}$$

Although this classifier improves sequence labelling when compared to HMM it suffers from label bias problem [43]. As its normalization is made locally, the classifier will tend to label the observations with states that have fewer path options. Figure 2.4 shows an example of sequence labelling being the states the classes and the arrows the paths with their transition probability. Following local transition probabilities the sequence classification would be State 1 → State 2 → State 2 → State 2. However, the probability of the sequence classification State 1 → State 1 → State 1 → State 1 is higher than previous.

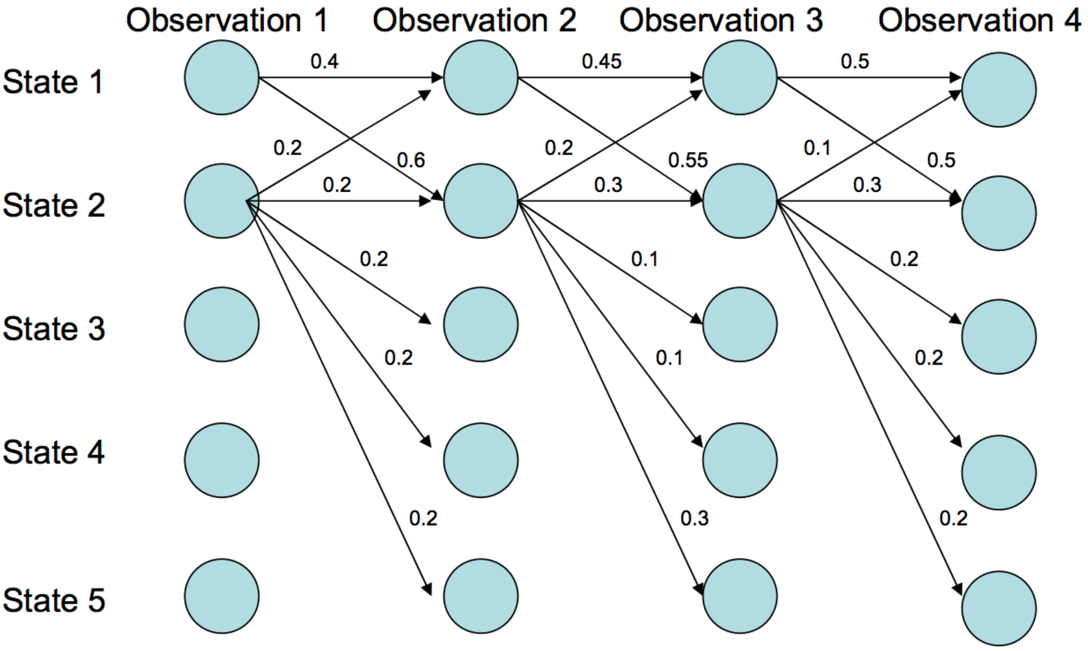


Figure 2.4: Example of label bias problem using MEMM. Image extracted from `https://cocoxu.github.io/courses/5525_slides_spring17/17_crf.pdf`.

Finally, CRF is a sequential supervised algorithm that enables automatic labelling of sequences based on undirected graphical models [43]. This algorithm is different from MEMM since its normalization is done globally instead of locally which prevents the label bias.

During training, first the algorithm builds all the possible combinations between the features and the output classes. Then, the algorithm learns which feature functions increase the accuracy of the predictions turning off the functions which decrease the performance. During this process, it also learns their weights. This way of training is also used on the previous algorithms, ME and MEMM.

$$P(\vec{y}|\vec{x}) = \frac{\exp \sum_{i=1}^{n} \sum_{j} \lambda_j f_j(\vec{x}, i, y_{i-1}, y_i)}{\sum_{y' \in Y} \exp \sum_{i=1}^{n} \sum_{j} \lambda_j f_j(\vec{x}, i, y'_{i-1}, y'_i)} \tag{2.6}$$

During the testing phase, the model maximizes the conditional probability of the output labels given the input features (equation 2.6), which is very similar to the equations of ME and MEMM. The naïve method would be testing all the tag combinations, which is impossible for classifications with many different tags, because this approach has an exponential growth (a document with m tokens and k different tags would have k^m different labels). That is why dynamic programming is important during this phase. The Viberti algorithm [44] allows to find the most probable label without testing all the combinations.

2.4 Vector Semantics

Vector semantics is a theory of meaning based on the representation of words in numerical vectors, i.e. to its Word Embedding (WE). Word representations are learned unsupervisedly by their distribution in large quantities of texts, which can be seen as an alternative to the manual creation of lexical knowledge bases or ontologies. In theory, the larger the amount of text and different words used, the better learned embeddings will capture the meaning of words.

The embedding vectors allow to check if different words are semantically related. Since the WE models are trained with various texts, words with similar meaning will appear in similar contexts e.g. abbreviations (the word “EEG” and the word “electroencephalogram” are similar words), synonyms (the words “abdomen” and “belly”) or words that belongs to the same context (“brain” and “skull”). Therefore their embedding representations will be closer in the WE dimensional map. In order to analyse if two words are related it is necessary to calculate their similarity , given by the cosine of their vectors, which is between 0 and 1. The closer the cosine is to 1, the more related the words are. There are different algorithms for learning WEs e.g. Latent semantic analysis [45], Word2Vec [46], GloVe [47] and FastText [48]. Among these algorithms, FastText is the only one which is able to capture morphology information. It is based on Word2Vec but instead of learning the words it learns their characters n-grams. Therefore it is able to learn not only the context but also the morphology which is useful for processing complex language grammars as Portuguese. We used this algorithm in our project as morphology information is important in clinical domain to help classifying some words with the proper class, e.g. words which end in “oma” have a high probability of being a tumor name (“melanoma”, “retinoblastoma”). Therefore, we will only give an explanation how FastText and Word2Vec algorithms work.

Word2Vec is a machine learning algorithm that produces vector representations for the words using a two-layer neural network. Given a collection of texts, word2vec can be trained using two different architectures, the Continuous Bag-of-Words (CBOW) or the Skip-gram.

CBOW architecture presented in figure 2.5 is trained by fitting the hidden layer weights in order to predict the current word based on the neighbour words while, the Skip-gram architecture does the inverse. As Skip-gram algorithm predicts the context of the a word based on the input word, it is better for smaller input collections of texts where rare words do not appear frequently. However, it requires more training time and it is worse than CBOW for the frequent words [46, 49].

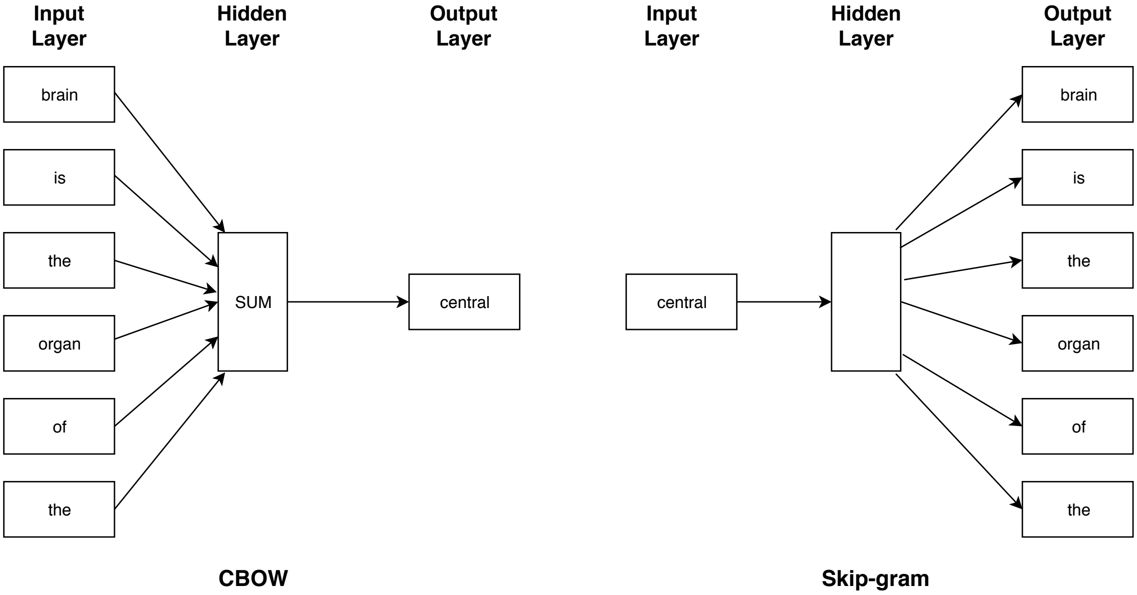


Figure 2.5: Skip-gram and Continuous Bag-of-words architectures for the Word2Vec algorithm using the following sentence "...brain is the central organ of the...". Figure adapted from figure 1 of [46].

FastText [48] is very similar to Word2Vec during its training. However, it considers all the current and context words as a sum of their character n-grams instead of atomic entities when fitting the model, e.g. taking the word "brain" and assuming that n=3 the word is represented by the following n-grams: "<br", "bra", "rai", "ain", "in>" and "<brain>". The "<" and ">" symbols are added to identify the beginning and the end of the word. Since this approach generates embeddings for each n-gram it requires more training time than Word2Vec. However, with this approach the WE model generates better word embeddings for rare words than Word2Vec, since it is able to learn not only the word semantics but also the word morphology. It is also able to generate vector representations for words which have never been seen by the model, i.e., out-of-vocabulary words.

2.5 Deep Learning Architectures

Deep learning architectures are machine learning models based on artificial neural networks that were inspired on the biological processes of the human brain [50, 51]. Figure 2.6 puts these models in a hierarchy that also includes Machine Learning and AI.

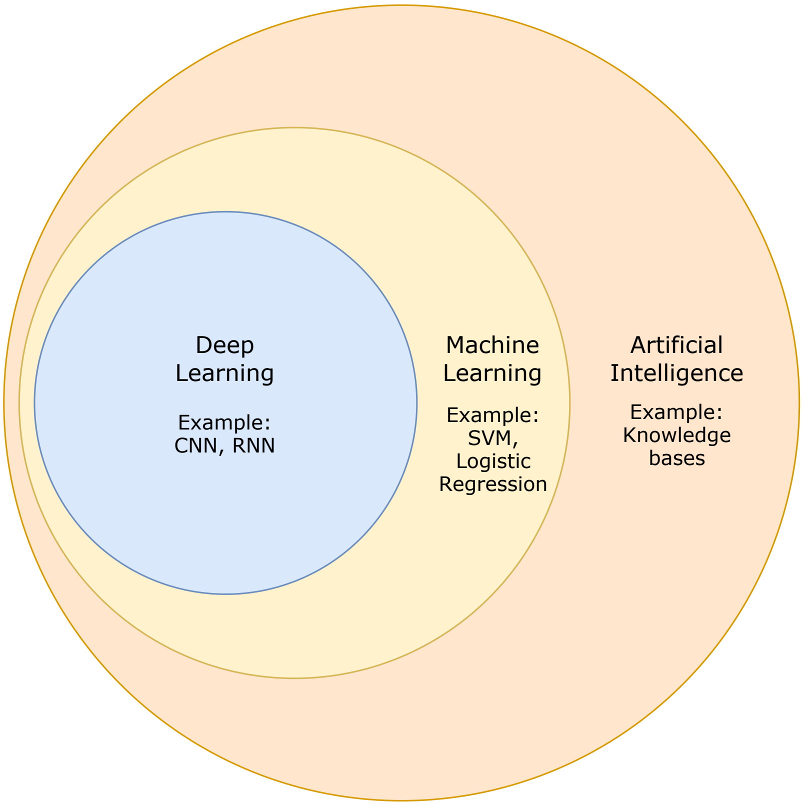


Figure 2.6: Venn Diagram representing the hierarchy of AI, machine learning and deep learning. It was adapted from figure 1.4 of [52].

They have recently become popular due to the increase of the amount of data and also as a result of the upgrade of the technology in terms of hardware and software [52]. There are several types of deep learning architectures such as Feed-Forward Networks (FFNs), CNNs, RNNs and Deep Belief Networks, which have been used in several domains as bioinformatics [53, 54], speech recognition [55, 56], image classification [57, 58], NLP, among others.

RNN is a popular architecture in IE. It allows to keep an internal memory that is maintained over the classification of all the sequence instances, i.e., it is able to use past information to predict the present and it is also able to use the future to correct the past predictions [59]. There are different types of RNN architectures such as the ones based on simple RNN layers, Long Short-Term Memory (LSTM) layers and Gated Recurrent Unit (GRU) layers. We used LSTM layers as they are more complex than GRU which could be useful for learning stronger relations between the tokens.

As stated in the last paragraph, RNN architecture with simple RNN layers can use information about the past to predict the present by feeding the next RNN unit with the output of the last unit, as presented in figure 2.7.

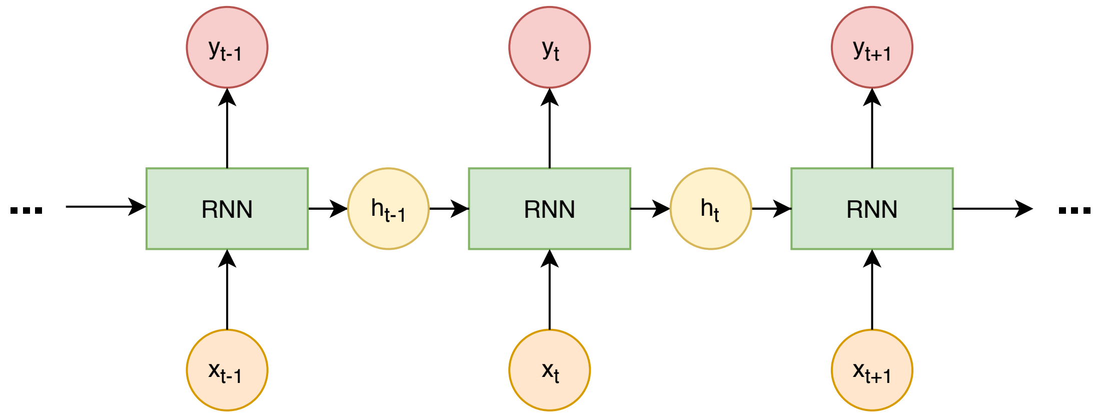


Figure 2.7: Example of simple RNN architecture. Adapted from figure 5 of [60].

However, this architecture suffers from vanishing gradient and exploding gradient during its training when the gradients that are responsible for changing the weights are being propagated [61]. Vanishing gradient occurs when the value of the gradients are smaller than one on the first layers of the architecture. Since several multiplications are made along the architecture as the values of the gradients are smaller than one they will tend to zero and cause no variation in the weights of the last layers. Exploding gradient occurs when the norm of the gradients get larger until they reach too large values which crash the training of the model.

Besides it has a long term dependency problem since the information learned from past samples of the sequence does not disappear [62]. In order to overcome such disadvantages, Hochreiter and Schmidhuber presented LSTM [63], a type of RNN that is able to control the flow of the information over the time steps (e.g. over the words of a text). The cell state, C, works as a memory, in other words, it keeps the old information. However, this memory is controlled by the forget gate, f. If the forget gate outputs a vector of zeros the multiplication with the old cell state will be zero and consequently the memory will be erased. On the other hand, if the output of this gate is a vector of ones, all the old information flows through the cell. Also, the input gate, i, controls how much information goes to the cell state in each time step. Supposing that the result of this gate is a vector of numbers which are near zero, almost no information is passed to the new cell state. The output gate, o, controls how much information from the hidden, h, and input states, x, is used to compute the new hidden state. The σ function used in all these gates is the sigmoid function which maintains the output of each gates in a range between 0 and 1. All the calculations made in the LSTM unit (figure 2.8) are presented in equations 2.7, 2.8, 2.9, 2.10 and 2.11.

During training, the network learns the best weight matrixes W. There are two types of matrixes W, the ones that multiply with the input vectors, W^x, and the ones that multiply with the hidden data W^h. Both matrixes are responsible for giving more importance to certain features rather than others. However, the first ones are also responsible for converting the dimension of the input data on the dimension of the hidden data. After this conversion the LSTM unit is able to sum both vectors and input it through the three gates. The model also learns the best bias vector, b.

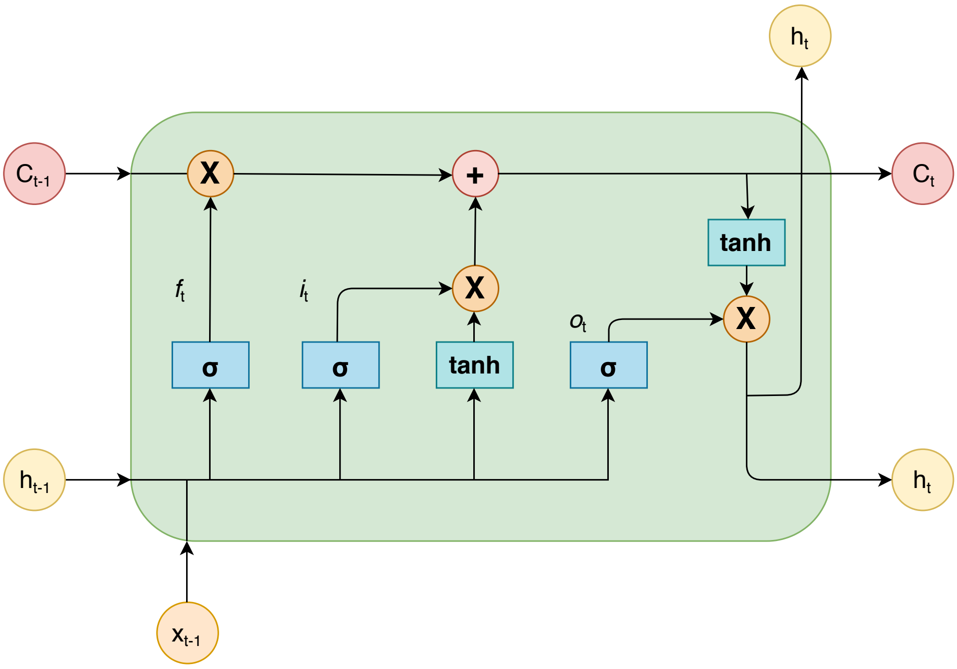


Figure 2.8: LSTM unit. The forget gate, input gate, output gate and cell states are represented by f_t, i_t, o_t, C_{t-1}, C_t, respectively. It was adapted from figure 6 of `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`.

$$f^{(t)} = \sigma(W^{fx} \cdot x^{(t)} + W^{fh} \cdot h^{(t-1)} + b_f) \tag{2.7}$$

$$i^{(t)} = \sigma(W^{ix} \cdot x^{(t)} + W^{ih} \cdot h^{(t-1)} + b_i) \tag{2.8}$$

$$o^{(t)} = \sigma(W^{ox} \cdot x^{(t)} + W^{oh} \cdot h^{(t-1)} + b_o) \tag{2.9}$$

$$C^{(t)} = f^{(t)} \cdot x^{(t)} + i^{(t)} \cdot tanh(W^{x} \cdot x^{(t)} + W^{h} \cdot h^{(t-1)} + b) \tag{2.10}$$

$$h^{(t)} = tanh(C^{(t)}) \cdot o^{(t)} \tag{2.11}$$

2.6 Evaluation

In order to measure progress and check whether our model is getting good performances, evaluation metrics have to be adopted. In this work, we have used K-Fold Cross Validation (CV) to train and validate our models, with Recall, Precision and F1-Score to evaluate them. We also split the evaluations in relaxed and strict evaluations. Relaxed or one-point performance measures the performance of the model for each token, while the strict performance considers all occurrences, i.e., one occurrence is well predicted if all its tokens are well predicted too. For example, with the relaxed evaluation, "síndrome depressiva" (*depressive syndrome*) counts as two tokens, i.e, each token's tag is independently compared to its golden tag. With the strict evaluation, if the model fails on a single token's tag, all NE occurrence is considered incorrect.

CV is a method of validation that splits all the dataset in smaller groups. The number of groups is decided *a priori* by who is checking the model performance. These groups are usually called folds that is why K-Fold CV is frequently associated with the number of folds e.g. K-Fold CV with a K=10 means that the dataset is splited in 10 different groups where each group has 90% for training and 10% for validation as shown in figure 2.9.

This method of validation is more robust than a holdout validation where the dataset is just splitted in e.g. 70% for training and 30% for validation, possibly resulting in a not representative corpus.

Recall, also known as binary decisions sensitivity, measures the fraction of relevant samples that have been retrieved over the total amount of relevant samples, while precision measures the percentage of exactness, i.e., it gives a ratio between the number of instances labelled with one class over all the instances classified with the

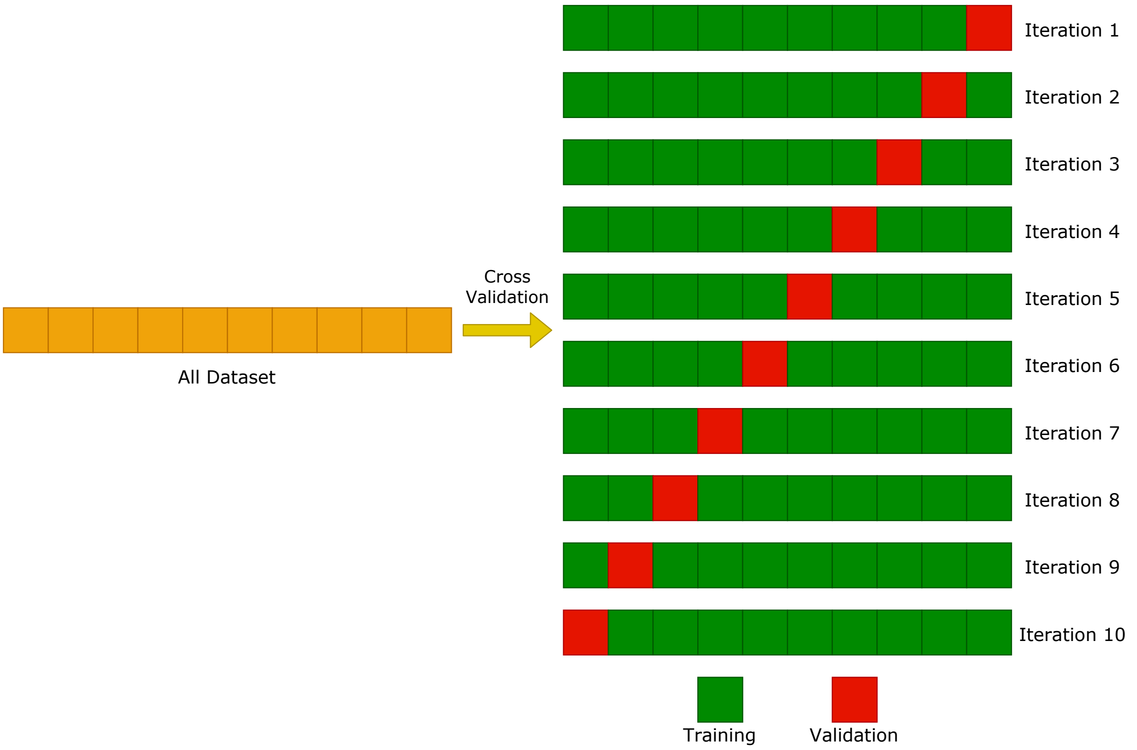


Figure 2.9: Example of a K-Fold CV using a K=10.

same class. Both measures are presented in equations 2.13 and 2.12, respectively. F1-Score is the harmonic average between recall and precision. A good F1-Score means that the model has a low number of false positives and false negatives. It is usually used for analysing the performance of a model on a dataset with a unbalanced class distribution and is computed according to equation 2.14.

$$Precision = \frac{True\, Positive\, (TP)}{True\, Positive\, (TP) + False\, Positive\, (FP)} \tag{2.12}$$

$$Recall = \frac{True\, Positive\, (TP)}{True\, Positive\, (TP) + False\, Negative\, (FN)} \tag{2.13}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.14}$$

As we had a multiclass dataset, we got different performance measures for each class. In order to check the performance of the model in all classes, we have to make an average of all the performances. For that, we can use a micro average, macro average or weighted average, in order to have a better insight of the results. The micro average is calculated taking into account the contribution of each class, while the macro average treats all the classes equally. Finally, the weighted average presented is calculated taking into account the weight of each class, i.e., the classes with more samples contribute with a higher value for the average value. The equations used to calculate all these three averages for precision are presented in equations 2.16, 2.18, 2.20 while for recall the equations are 2.15, 2.17, 2.19. The equations for recall micro and weighted averages are the same because the number of instances of each class is equal to the sum of true positives and false negatives.

$$Recall\,Micro-Average=\frac{\sum_{i=1}^{number\,of\,classes}TP_i}{\sum_{i=1}^{number\,of\,classes}TP_i+FN_i} \tag{2.15}$$

$$Precision\,Micro-Average=\frac{\sum_{i=1}^{number\,of\,classes}TP_i}{\sum_{i=1}^{number\,of\,classes}TP_i+FP_i} \tag{2.16}$$

$$Recall\,Macro-Average=\frac{\sum_{i=1}^{number\,of\,classes}Recall_i}{number\,of\,classes} \tag{2.17}$$

$$Precision\,Macro-Average=\frac{\sum_{i=1}^{number\,of\,classes}Precision_i}{number\,of\,classes} \tag{2.18}$$

$$Recall\,Weighted-Average=\sum_{i=1}^{number\,of\,classes}\frac{TP_i}{TP_i+FN_i}\times\frac{TP_i+FN_i}{Total\,Samples} \tag{2.19}$$

$$Precision\,Weighted-Average=\sum_{i=1}^{number\,of\,classes}\frac{TP_i}{TP_i+FP_i}\times\frac{TP_i+FN_i}{Total\,Samples} \tag{2.20}$$

3

Related Work

This chapter reviews the previous and current state of the art related to Named Entity Recognition (NER). It starts by presenting some approaches for dataset labelling. Then, it presents NER models beginning with the ones based on rules and terminologies and then those based on machine learning and deep learning algorithms. Furthermore, we present how other authors annotated their datasets and the comparison between in-domain and out-of-domain Word Embeddings (WEs) models as well. The chapter ends with two tables that summarize the models used in the studies presented throughout this section.

3.1 Dataset Annotation

Building a model for clinical NER requires access to much clinical textual data. Although much text of this kind is produced everyday, its availability is highly limited due to strict ethical regulations that constrain using data with personal information, as in clinical case or diagnostic test reports. Still, when available, such texts constitute valuable sources of data, and may be used in the development of models for Information Extraction (IE), including NER. So that systems learn how to annotate Named Entities (NEs), the latter have to be annotated on a subset of texts, which can be used as training and/or testing data. Studies that present dataset labelling include Uzuner *et al.*[64], who annotated 871 medical records with Medical Problems, Treatments and Tests, in order to provide a dataset for the 2010 i2b2/VA concept extraction shared task; or Stubbs *et al.*[65], who labelled 1,304 individual longitudinal records with heart-risk NEs (e.g. Diabetes references or Hypertension) with 0.95 of Agreement Ratio (AR). Beyond English, some studies involved the creation of datasets in other languages. Skeppstedt *et al.* [66] annotated Disorders, Findings, Body Structures and Pharmaceutical Drugs, in 1,104 clinical notes in Swedish, with agreement ratios of 0.79, 0.66, 0.80 and 0.90, respectively. Mykowiecka *et al.* [30]

annotated 700 mammography reports and 100 diabetic discharge documents, in Polish, with NEs that carry information about Pathological Findings, Breast Tissue, and Crucial Health information about diabetic patients. Ferreira *et al.* [10] manually labelled 90 clinical notes in Portuguese with NE classes such as Condition, Anatomical Site and Finding. Although made for Portuguese, this dataset is not available due to ethical regulations, only the annotation guidelines followed.

3.2 Rules and Dictionary-based Models

NER has been tackled with rules based on regular expressions combined with the exploitation of medical vocabularies [31] or ontologies [29, 10]. Skeppstedt *et al.* [29] assigned NE classes to tokens based on their presence in terminologies. The authors made 11 preprocessing experiments. On their baseline model, first, they created a database which have terms that belong to disorders, findings and body structures. Then, they split the documents into several sentences in order to compare them with the terms presented in the database. If the database contains a term which is equal to the sentence all its tokens are annotated with the respective NE class, e.g., if the sentence matches a term that belongs to disorders all the tokens are labelled as disorders. Thereafter, the sentence is split into several tokens and each one of these tokens are compared with the terms of the database. Once again, if any token is equal to any term it is labelled with the respective NE class. Finally, if a token is assigned to more than one class, some classes are preferred over others (e.g., body structures are preferred over disorders and findings). Skeppstedt *et al.* [29] assigned NE classes to tokens based on their presence in terminologies. If a token is assigned to more than one class, some classes are preferred over others (e.g., body structures are preferred over disorders and findings). Gold *et al.* [31] resorted to RxNorm[1], a standard vocabulary for names of clinical drugs. Mykowiecka *et al.* [30] handcrafted a set of rules and, despite achieving precision and recall above 80%, admitted that the rules are highly dependent on the quality of the reports. Ferreira *et al.* [10] also exploits terminologies and reports results near 100% for most entity classes, but their model was only assessed on ten discharge letters.

Despite the high performances reported, the development of rule-based models is time-consuming due the exhaustive labour involved in the creation of rules and dictionaries, not to mention that these models are generally tuned for the target documents, thus making adaptation to other types of text difficult. Classification

[1] `https://www.nlm.nih.gov/research/umls/rxnorm/`

can also be quite slow, because it has to look up on dictionaries or compare with pre-designed templates for finding what class each token belongs to. Furthermore, some words may appear in different entities (e.g., “tumor” can be labelled as a condition or a result of a diagnostic test; “lumbar puncture” can be labelled as a diagnostic test or a therapy) which may not be distinguishable for models based on rules and dictionaries.

3.3 Machine Learning Approaches

To make the development of NER models more robust, it is fundamental to adopt machine learning algorithms, where the computer learns how to predict the class of an entity based on a set of annotated examples and extracted features. These algorithms could be separated by depth being the shallow ones the more classical methods and the deep learning approaches the more recent ones.

3.3.1 Shallow Machine Learning Algorithms

In this scope, NER is typically seen as a sequence-labelling task, and models for this purpose are often applied, e.g., Hidden Markov Models (HMM), Maximum Entropy Markov Models (MEMM), Conditional Random Fields (CRF), Support Vector Machines (SVM) and Decision Trees. Yet, literature consistently suggests that among these algorithms the best results are obtained with CRF [67, 68].

CRF models have been trained on NER from clinical text [66] and outperformed rule-based systems in a similar scenario [29]. Relevant features for this purpose have been analysed [66] and included dictionary forms (lemmas) of the current and previous tokens; the Part-of-Speech (POS) tag of the current, following, and the two previous tokens; the terminology matching class for the current and the previous tokens; the compound splitting features for the current token; and the orthographic features for the current token.

Results of classic CRF have also been improved by exploiting distributional semantic features learned from large clinical corpora.

Prototypical representations of each NE class were learned by exploiting NE annotations and a distributional model based on random indexing [32]. Different types of WE have also been learned for this purpose [33]: using word vectors of real numbers (e.g. ventricle: [0.194, -1.492, 2.407, 0.996, 0.379, 2.384, -1.808, -0.608, ...]), discrete values derived from the previous ones (e.g. atrium: [0, -, +, +, +, +, -, -, ...]), and based on a matrix with the prototypical words for each class based on Normalized Pointwise Mutual Information (NPMI) (e.g. warfarin: [coumadin, lisinopril, metoprolol, protonix, aspirin, colace, heparin, tylenol, percocet, ...]). NPMI is the normalized form of Pointwise Mutual Information. It represents the association between two variables x (e.g. feature) and y (e.g class), i.e., it is the ratio of the probability of their occurrence at the same time and the product of their independent probabilities.

$$nPMI(label,word) = \ln \frac{p(label,word)}{p(label)p(word)} \times \frac{1}{-\ln p(label,word)} \tag{3.1}$$

3.3.2 Deep Learning Approaches

In recent years, deep learning approaches have been used for NER, leading to state of the art results. Clinical NER is not an exception, with such models used for extracting data from Electronic Medical Record (EMR).

Adopted architectures include Recurrent Neural Networks (RNNs), with simple RNN layers, Long Short-Term Memory (LSTM) layers, Bidirectional Long Short-Term Memory (BiLSTM) layers or Gated Recurrent Unit (GRU) layers; Convolutional Neural Network (CNN); and also Feed-Forward Network (FFN). Luu *et al.* [69] showed that a vanilla RNN outperforms a FFN using the same features on clinical texts provided in the CLEF eHealth 2016 task [70] on the extraction of relevant information from nursing shift changes notes. This was expected because FFNs do not consider past information.

Chokwijiktul *et al.* [36] evaluated the performance of CNN, RNN, LSTM, BiLSTM and GRU networks in identifying heart risk factors in EMRs and found that BiLSTM networks achieved the best F-measure. Furthermore, they show that such models perform near the rule-based and shallow machine learning models, but without resorting to gazetteers or other knowledge bases. Additionally, Wu *et al.* [35] compared different classifiers such as a CRF, a CNN and a BiLSTM network for NER, using the dataset of the 2010 i2b2 Natural Language Processing (NLP) challenge. They also compared their models with the best model at the time (Structured SVM) and the best model trained during the competition (Semi-Markov model). They used pre-trained WEs as features for the BiLSTM network and the CNN. For the CRF, they used three different feature sets: only word and n-gram features; the previous plus linguistic features and document level features, such as section names; and finally, all the previous plus features from general clinical NLP systems (MedLEE, MetaMap and KnowledgeMap) and gazetteer features from the UMLS terminology. Similarly to Chokwijiktul *et al.* [36], they report that the BiLSTM network outperformed all the others.

Others developed a BiLSTM network with a character embedding layer, a WE layer and a CRF layer. Xu *et al.* [71] evaluated their architecture on the NCBI Disease Corpus (793 PubMed medical literature abstracts), while Jauregi Unanue *et al.* [72] evaluated their models with three different datasets (2010 i2b2/VA dataset, DrugBank and MedLine). Both showed that the CRF layer and the character embedding feature have great importance on the performance of a BiLSTM network.

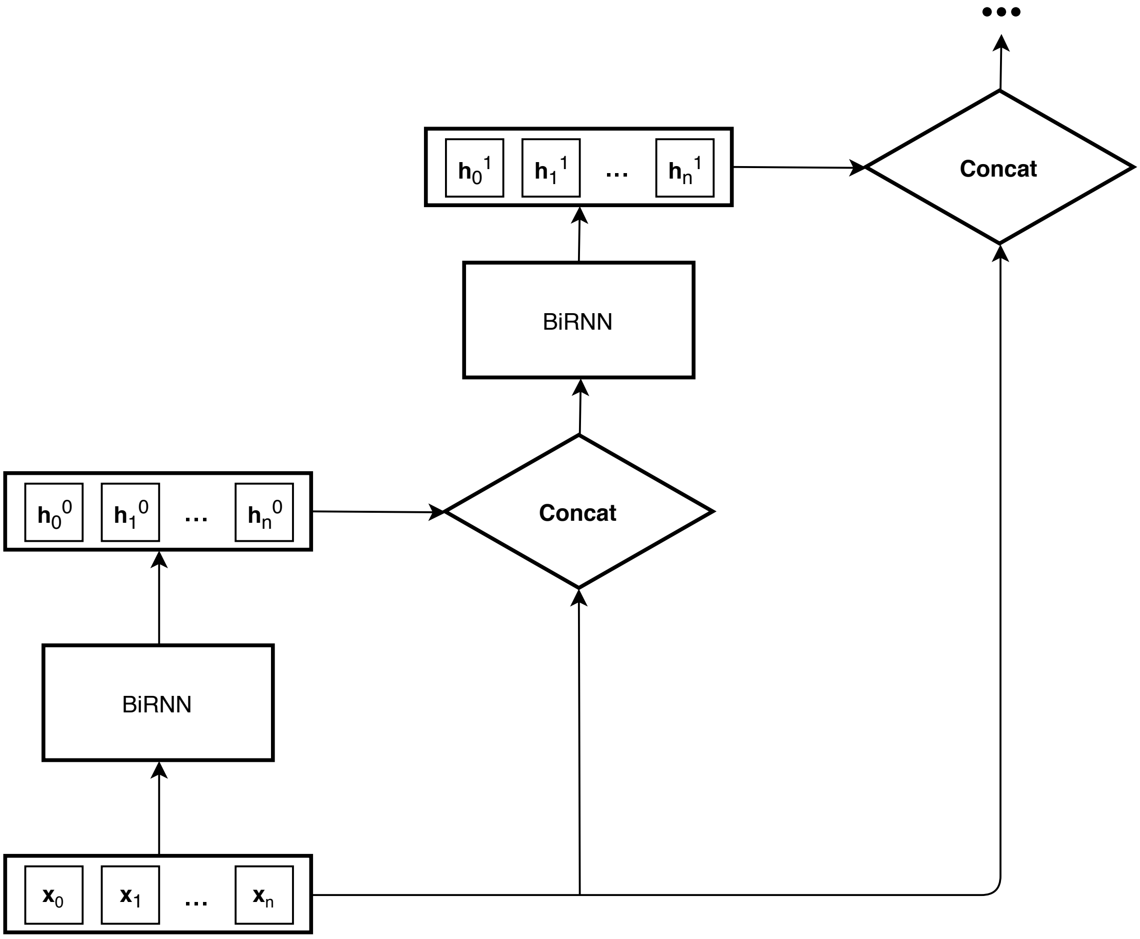


Figure 3.1: Residual learning architecture proposed by Tran *et al.*. The figure was exported from [73].

Recently, Tran *et al.* [73] and Prakash *et al.* [74] have presented a new approach for NER and paraphrase generation, respectively. They presented a stacked RNN model based on residual learning, an approach based on the architecture presented by He *et al.* [75] for image classification. According to He *et al.* [75], this approach allows to correct the degradation problem of the deeper neural networks that happens with the increase of the depth when the training accuracy saturates and then it degrades quickly. Although the three works used residual learning, they used it in a different way. In [75] and [74] the residual connections were made summing the input data with the output of each layer while in [73] the connections were made concatenating the input data with the output data of each layer. According to Tran *et al.* [73] this approach was adopted due to it was not necessary to perform dimensionality reduction as in summing approaches.

Although these models became the trend in NER, they rely heavily on the quality of the WE models for converting each word to its embedding vector. On the clinical domain, Griffis *et al.* [76] compared WEs using in-domain and out-of-domain corpora. In-domain corpora were made by two different datasets, one with 154,967 EMRs and a subset with 17,952 EMRs documents focused on Physical Therapy and Occupational Therapy. Out-of-domain corpora comprised 14.7 million abstracts from the 2016 PubMed baseline and two million free-text documents released as part of the MIMIC-III critical care DB. Besides those, they used a FastText model, pre-trained on Wikipedia 2017 documents. They reported that, with WEs trained with small in-domain corpora, results were similar to those achieved with the large out-of-domain corpora. Jauregi Unanue *et al.* [72] additionally showed that re-training WE models with domain-specific texts improves the performance of the model.

3.4 Related Work in Portuguese

As previously discussed in section 3.2, Ferreira et al. [10] performed clinical NER on Portuguese clinical discharge letters. However, their model was based on rules and terminologies instead of machine learning algorithms.

Although not on the clinical domain, there is some related work on Portuguese. On general NER, de Castro *et al.* [77] recently achieved state-of-art results using a BiLSTM-CRF model. On distributional similarity, Hartmann *et al.* [78] compared Portuguese WEs, learned with different methods, in both intrinsic (syntactic and semantic analogies) and extrinsic (POS and sentence similarity) tasks. There are also studies suggesting that, in tasks such as POS and NER, combining character embedding with pre-trained WEs outperforms approaches that use only WEs [79, 80].

3.5 Summary

Table 3.1 presents a summary of the clinical NER studies discussed throughout this chapter. It contains the methods used by the authors, the amount of texts as well as the number of tokens, their type (e.g. discharge letters or nursing notes), their language (e.g. Portuguese, English or Swedish) and the extracted information. 3.2 presents the results for each work shown on table 3.1. Despite of the presented results were obtained with similar methodologies to those used by us in this project, they are not directly comparable due to the difference between the datasets (texts and language).

Table 3.1: Summary of the methods, dataset language and extracted information of clinical NER related work models.

References	Method	Language	Texts	Extracted Information
Mykowiecka et al. [30] 2009	Rules and Gazetteers	Polish	All Dataset: 2439 Mammography reports and 606 Hospital records of diabetic patients; Test Dataset: 705 Mammography reports and 100 Hospital records of diabetic patients.	Tissue features and location, patient data, diabetes features and test results
Skeppstedt et al. [29] 2012	Rules and Terminologies	Swedish	Part of the Stockholm EPR Corpus [81] (26,011 tokens)	Body Structure, Disorder and Finding
Ferreira et al. [10] 2010	Rules and Terminologies	Portuguese	All Dataset: 915 Discharge letters (51695 tokens); Annotated Dataset: 90 Discharge letters; Test Dataset: 10 Discharge letters	Condition, Anatomical Site, Evolution, Examination, Finding, Location, Therapeutic, DateTime and Value.
Gold et al. [31] 2008	Rules and Terminologies	English	26 Discharge summaries	Medication events such as dosage information and route administration
Skeppstedt et al. [66] 2014	CRF	Swedish	All Dataset: 1,148 texts from Stockholm EPR Corpus [81] (70,852 tokens); Development Dataset: 45,482 tokens; Test Dataset: 25,370 tokens.	Disorder, Finding, Drug, Body Structure and Disorder+Finding

Continues on next page

References	Method	Language	Texts	Extracted Information
Henriksson et al. [32] 2014	CRF	Swedish	100 clinical texts from Stockholm EPR PHI Corpus and 10 million unannotated clinical notes	Protected health information
Wu et al. [33] 2015	CRF	English	Discharge progress texts from i2b2 2010 (349 for training and 477 for test) and discharge, radiology, ECG and ECHO notes from SemEval 2014 (298 for training and 133 for test) and 403,871 unannotated discharge, radiology, ECG and ECHO notes from MIMIC II	Problem, Test and Treatment
Luu et al. [69] 2018	FFN and RNN	English	200 Nursing shift-change handover texts (100 for training and 100 for test)	Appointment/ Procedure, future, medication, my shift and patient introduction
Chokwijiktul et al. [36] 2018	CNN, RNN, LSTM, BiLSTM and GRU	English	1,304 medical records from 2014 i2b2/UTHealth shared task (790 for training and 514 for test)	Heart disease risk factors
Wu et al. [35] 2018	CNN, BiLSTM and three baseline CRF models	English	Discharge progress texts from i2b2 2010 (349 for training and 477 for test) and 403,871 unannotated discharge, radiology, ECG and ECHO notes from MIMIC II	Problem, Treatment and Test
Xu et al. [71] 2017	BiLSTM-CRF with character embeddings	English	793 PubMed abstracts from NCBI Disease Corpus (593 for training, 100 for development and 100 for test)	Diseases
Jauregi Unanue et al. [72] 2017	BiLSTM-CRF with character embeddings	English	Clinical texts from i2b2/VA (170 for training and 256 for test), from DrugBank (730 for training and 54 for test) and from MedLine (175 for training and 58 for test) and 53,423 unannotated distinct hospital admissions from MIMIC-III	Drug names, problems and tests
Griffis et al. [76] 2018	BiLSTM-CRF with character embeddings	English	250 deidentified EMR documents and 154,967 unannotated EMR documents and 17,952 Physical Therapy and Occupational Therapy unannotated documents	Descriptions of mobility status, Measurement scales related to mobility activity

Table 3.2: Summary of the results of clinical NER related work models.

References	Precision	Recall	F1-Score
Mykowiecka et al. [30] 2009	Most of the evaluated templates: Above 80%	Most of the evaluated templates: Above 80%	-
Skeppstedt et al. [29] 2012	Body Structure: 74%, Disorder: 75% and Finding: 57%	80%, 55% and 30%	-

Continues on next page

References	Best Precision	Best Recall	Best F1-Score
Ferreira et al. [10] 2010	Condition: 93%, Anatomical Site: 100%, Evolution: 100%, Examination: 69%, Finding: 93%, Location: 100%, Therapeutic: 99%, DateTime: 100% and Value: 100%	-	-
Gold et al. [31] 2008	94.1%	82.5%	-
Skeppstedt et al. [66] 2014	Disorder: 80%, Finding: 72%, Drug: 95%, Body Structure: 88% and Disorder+Finding: 80%	82%, 65%, 83%, 82% and 76%	81%, 69%, 88%, 85% and 78%
Henriksson et al. [32] 2014	92.1%	81.3%	85.5%
Wu et al. [33] 2015	i2b2 2010: 85.2% and SemEval: 78.9%	80.6% and 78%	82.8% and 78.1%
Luu et al. [69] 2018	FFN: 44.5% and RNN: 71.8%	27.4% and 62.6%	33.9% and 66.7%
Chokwijiktul et al. [36] 2018	CNN: 83.83%, RNN: 88.44%, LSTM: 88.36%, BiLSTM: 89.83% and GRU: 90.02%	92.45%, 89.56%, 91.91%, 91.80% and 90.91%	87.93%, 89.00%, 90.10%, 90.81% and 90.46%
Wu et al. [35] 2018	CNN: 84.91%, RNN: 85.33% and CRF: (82.32%, 83.25% and 86.52%)	80.73%, 86.56% and (72.92%, 76.75% and 81.04%)	82.77%, 85.94% and (77.33%, 79.87% and 83.60%)
Xu et al. [71] 2017	84.80%	76.12%	80.22%
Jauregi Unanue et al. [72] 2017	i2b2/VA (problem): 81.29%, i2b2/VA (test): 84.74%, DrugBank (group): 81.69%, DrugBank (drug): 94.77%, MedLine (group): 69.14%, MedLine (drug): 73.89%	83.62%, 85.01%, 87.88%, 89.56%, 60.22% and 77.33%	82.44%, 84.87%, 84.67%, 91.83%, 64.37% and 75.57%
Griffis et al. [76] 2018	Mobility (Strict evaluation): 71.9% MIMIC - Word2Vec, Mobility (Relaxed evaluation): 86.0% PubMed - FastText, ScoreDefinition (Strict evaluation): 93.6% PubMed - FastText and ScoreDefinition (Relaxed evaluation): 98.1% PubMed - FastText	65.9% PubMed - FastText, 87.7% PubMed - Word2Vec, 95.8% BTRIS/Physical Therapy-Occupational Therapy Reports - FastText and 99.9% BTRIS - FastText	68.2% MIMIC - Word2Vec, 83.6% MIMIC - Word2Vec, 93.9% Physical Therapy-Occupational Therapy Reports - FastText and 98.7% PubMed - Word2Vec

Most of the clinical NER studies identified were for English, despite a minority for other languages, such as Polish [30], Swedish [29, 66, 32], and Portuguese [10]. Although the latter is our target language, their approach is mostly rule-based and the authors did not make available their dataset due to privacy legislation.

4

Experimental Setup

This chapter starts with a description of the used textual data, its preprocessing and guidelines followed by its annotation. It also describes the resulting dataset with some numbers on its contents and revision. Then, it explains how the used Word Embedding (WE) models were trained and how the grid search and feature selection for the Conditional Random Fields (CRF) model were performed. It ends by explaining the architecture of the deep learning Named Entity Recognition (NER) models, including how their hyperparameters grid search was done.

4.1 Dataset

Three different datasets were used in different stages of this work:

- For training and validation, 281 clinical case texts collected from numbers 1 and 2 of volume 17 of the clinical journal Sinapse [82, 83], published by the Portuguese Society of Neurology.
- For testing, a small set of 20 clinical texts obtained from the Neurology service of the Coimbra Hospital and Universitary Centre (CHUC), in Coimbra, Portugal. These included admission notes, diagnostic test reports and patient discharge letters and were originally used in the development of the European Epilepsy Database [84], with data approved by the Ethics Committee of the different institutions involved in the database development (Freiburg, Ethics Commission of the Universitätsklinikum Freiburg; Paris, Ethics Commission of the Hôpital Universitaire Pitié-Salpêtrière; Coimbra, Ethics Commission of the Hospitais da Universidade de Coimbra).
- For training the in-domain WE model, a total of 3,377 clinical texts were collected from all the volumes of the Sinapse journal, published between 2001

and 2018[1]. Although the journal contains clinical cases and experimental reports, we just collected the clinical cases.

Since all the texts were in a raw format, they first had to be preprocessed. This was made with NLPPort [85], a set of automatic tools for Portuguese Natural Language Processing (NLP) available on GitHub[2]. More specifically, we used a tokenizer (TokPort) for splitting the text into tokens (e.g., words, punctuation); a Part-of-Speech (POS) tagger (TagPort), for assigning a POS to each token (e.g., noun, verb); and LemPort, for normalizing the word into its lemma form. NLPPort was used instead of Spacy[3], NLTK[4] or Standford NLP[5] because, although these can be trained for Portuguese, NLPPort targets Portuguese text specifically.

After preprocessing, the dataset was represented in the CoNLL-2003 format [86], a common format for textual data annotation, with a token per line and its attributes in the following columns, separated by tabs. We saved tokens in the first column, POS tags in the second, and lemmas in the third, as table 4.1 illustrates.

4.1.1 Annotation

As the datasets were collected in raw, it was necessary to label them with the Named Entity (NE) classes representing information that it is important to extract. These labels were to be added, manually, in the fourth column.

For that, our starting point was the same guide used by Ferreira [87], written by physicians and linguists for the annotation of clinical text. This guide covers the following NE categories: Anatomical Site, Condition, Characterization, DateTime, Evolution, Location, Negation, Results, Route of Administration, Test, Therapeutics and Values. Here is a brief description of each class:

- Anatomical Site represents all the references to anatomical locations, e.g. "pulmonar" (*pulmonary*) in "A AngioTC pulmonar..." (*The pulmonary AngioTC...*). It is usually next to Condition, Test or Results.
- Condition is related to the clinical signals, symptoms, diagnosed diseases and pathologies.

[1] `http://www.sinapse.pt/archive.php`
[2] `https://github.com/rikarudo/NLPPORT`
[3] `https://spacy.io/`
[4] `https://www.nltk.org/`
[5] `https://stanfordnlp.github.io/stanfordnlp/`

- The words that belong to Characterization modify the meaning of Condition instances, e.g. "suspeita" (*suspicion*) in "suspeita de Arterite de Takayasu" (*suspicion of Takayasu's arteritis*) or "significativo" (*significant*) in "shunt direito-esquerdo significativo" (*right-left significant Takayasu's arteritis*).

- DateTime expresses all the temporal references (dates, duration or frequencies).

- Evolution is the class for the clinical progression of the patient, e.g. "melhoria" (*improvement*) in "melhoria dos défices focais" (*improvement of the focal deficits*).

- Location represents geographicaly locations, e.g. "Coimbra" or "domicílio" (*at home*).

- Negation contains all the negation expressions found in the texts, e.g. "não" (*no*) in "Não foi encontrada etiologia" (*No etiology was found*) or "sem" (*without*) in "sem áreas de comprovada restrição" (*without proven restriction areas*). It is an important NE class since it allows to know if a result or condition is shown in a test.

- All the words related to results of any Test that are not annotated as a Condition belong to Results, e.g. "normal" (*normal*) in "exame citoquímico de LCR normal" (*cytochemical examination of cerebrospinal fluid was normal*) or "níveis baixos de coenzima q10" (*low levels of coenzyme q10*) in "biópsia muscular com níveis baixos de coenzima q10" (*muscle biopsy with low levels of coenzyme q10*).

- Route of Administration and Therapeutics are related to medication or therapies. Therapeutics contains the words related to all the processes (chemical, physical or surgical) which are done to cure the patient and the Route of Administration consists of the way chemical therapeutics are taken, e.g. "oral" (*oral*) in "prednisolona oral" (*oral prednisolone*).

- Finally, Values encloses all the numbers which are not related to dates, e.g. "7 células" (*7 cells*) in "Exame de LCR com 7 células" (*examination of cerebrospinal fluid with 7 cells*) or "5 mg" in "medicado com apixabano 5 mg" (*medicated with apixaba 5 mg*).

Although the guide has subclasses for almost all NE classes, e.g. a Test can be physical, analytic and imagiological, we did not annotate them in the dataset. Furthermore, minor adaptations were made, namely: (i) the Location NE class was not considered because it does not represent clinical information. Although DateTime is not directly related to clinical information but to temporal expressions, we considered that this one was important since it provides temporal information about the events written in the reports; (ii) the classes Genetics and Additional Observations were added. The former identifies information about genes (e.g., "...variante do *gene CoQ2* em ..." (*...gene CoQ2 variant in...*)), and the latter all the extra unlabelled information about the patient, such as references to family diseases or patient opinions, among others (e.g., "...retomou Dasatinib (*decisão do doente* e *hematologista assistente*), desenvolvendo..."). This resulted in 13 NE classes, as illustrated in table 4.4.

We further adopted the Inside-Outside-Beginning (IOB) format, which, as stated in section 2.2, allows to distinguish between tokens in the beginning and inside a NE. Therefore, the dataset considers 27 different tags, two for each NE class, plus the Out tag, for tokens not belonging to a NE. Table 4.1 illustrates the annotated data.

Table 4.1: Example of dataset annotation. Sentence: "...de 66 anos, com antecedentes de dislipidemia e síndrome depressiva, começou por..." (*...66 years old, with a history of dyslipidemia and depressive syndrome, began....*

Token	POS Tag	Lemma	IOB Tag
de	prp	de	O
66	num	66	O
anos	n	ano	O
,	punc	,	O
com	prp	com	O
antecedentes	n	antecedente	B-DT
de	prp	de	O
dislipidemia	n	dislipidemia	B-C
e	conj-c	e	O
síndrome	n	síndrome	B-C
depressiva	adj	depressivo	I-C
,	punc	,	O
começou	v-fin	começar	O
por	prp	por	O

Tables 4.2 and 4.4 provide a quantitative analysis of the training and validation datasets, while tables 4.3 and 4.5 a quantitative analysis of the independent test set. Tables 4.2 and 4.3 quantify the tokens for each IOB tag (NT), the number of distinct tokens (NDT), and their ratios (NTR, NDTR). Finally, tables 4.4 and

4.5 show the number of NE occurrences (O), the number of distinct NE occurrences (DO) and their ratios (OR, DOR). As the test set has only reports related to epilepsy, it does not have NE occurrences of the Genetics.

Table 4.2: Quantitative analysis of the training/validation Dataset.

IOB Tags	NT	NTR (%)	NDT	NDTR (%)	Examples	Examples (English)
B-AS	2,491	4.272	770	6.794	seio (B-AS)	venous
I-AS	2,510	4.305	599	5.285	venoso (I-AS)	sinous
B-C	3,884	6.662	1,074	9.476	paramnésia (B-C)	reduplicative
I-C	3,634	6.233	1,269	11.196	reduplicativa (I-C)	paramnesia
B-CH	1,043	1.789	503	4.438	mais (B-CH)	more
I-CH	576	0.988	358	3.159	marcado (I-CH)	marked
B-DT	1,516	2.600	280	2.470	18 (B-DT)	18
I-DT	2,495	4.279	378	3.335	semanas (I-DT)	weeks
B-EV	794	1.362	184	1.623	desenvolveu (B-EV)	gradually
I-EV	452	0.775	120	1.059	gradualmente (I-EV)	developed
B-G	61	0.105	15	0.132	gene (B-G)	EGFR
I-G	62	0.106	47	0.415	EGFR (I-G)	gene
B-N	768	1.317	46	0.406	não (B-N)	not
I-N	2	0.003	2	0.018	impedindo (I-N)	hindering
B-OBS	217	0.372	153	1.350	restantes (B-OBS)	remaining
I-OBS	227	0.389	144	1.271	irmãos (I-OBS)	siblings
B-R	1,767	3.031	589	5.197	VS (B-R)	increased
I-R	2,520	4.322	922	8.135	aumentada (I-R)	ESR
B-RA	71	0.122	14	0.124	intravenoso (B-RA)	intravenous
I-RA	0	0.000	0	0.000		
B-T	2,041	3.501	490	4.323	estudo (B-T)	cytogenetic
I-T	2,113	3.624	677	5.973	citogénico (I-T)	study
B-THER	894	1.533	384	3.388	correção (B-THER)	correction
I-THER	709	1.216	332	2.929	de (I-THER)	of
B-V	410	0.703	276	2.435	0.8 (B-V)	0.8
I-V	584	1.002	112	0.988	células (I-V)	cells
O	26,463	45.388	1,596	14.082	-	-
Total	58,304	100,000	11,334	100.000	-	-

Reference: NT: Number of Tokens; NTR: Number of Tokens Ratio; NDT: Number of Distinct Tokens; NDTR: Number of Distinct Tokens Ratio; CH: Characterization; T: Test; EV: Evolution; G: Genetics; AS: Anatomical Site; N: Negation; OBS: Additional Observations; C: Condition; R: Results; DT: DateTime; THER: Therapeutics; V: Value; RA: Route of Administration; O: Out

Table 4.3: Quantitative analysis of the test Dataset.

IOB Tag	NT	NTR (%)	NDT	NDTR (%)
B-AS	17	0.628	13	1.343
I-AS	12	0.444	8	0.826
B-C	99	3.660	48	4.959
I-C	109	4.030	58	5.992
B-CH	51	1.885	42	4.339
I-CH	48	1.774	33	3.409
B-DT	130	4.806	67	6.921
I-DT	194	7.172	96	9.917
B-EV	52	1.922	30	3.099
I-EV	12	0.444	10	1.033
B-G	0	0.000	0	0.000
I-G	0	0.000	0	0.000
B-N	33	1.220	7	0.723
I-N	0	0.000	0	0.000
B-OBS	47	1.738	26	2.686
I-OBS	58	2.144	35	3.616
B-R	19	0.702	16	1.653
I-R	14	0.518	13	1.343
B-RA	3	0.111	3	0.310
I-RA	0	0.000	0	0.000
B-T	66	2.440	36	3.719
I-T	36	1.331	28	2.893
B-THER	88	3.253	62	6.405
I-THER	59	2.181	37	3.822
B-V	38	1.405	29	2.996
I-V	62	2.292	18	1.860
O	1,458	53.900	253	26.136
Total	2,705	100	968	100

Reference: NT: Number of Tokens; NTR: Number of Tokens Ratio; NDT: Number of Distinct Tokens; NDTR: Number of Distinct Tokens Ratio; CH: Characterization; T: Test; EV: Evolution; G: Genetics; AS: Anatomical Site; N: Negation; OBS: Additional Observations; C: Condition; R: Results; DT: DateTime; THER: Therapeutics; V: Value; RA: Route of Administration; O: Out

Table 4.4: NE classes description for Training/Validation Dataset.

NE Class	O	OR (%)	DO	DOR (%)
Anatomical Site	2,488	15.59	1,412	16.14
Condition	3,887	24.35	2,203	25.18
Characterization	1,044	6.54	632	7.22
DateTime	1,519	9.52	883	10.09
Evolution	793	4.97	331	3.78
Genetics	63	0.39	50	0.57
Additional Observations	217	1.36	166	1.90
Negation	768	4.81	48	0.55
Results	1,766	11.06	1,090	12.46
Route of Administration	71	0.45	14	0.16
Test	2,041	12.79	1,012	11.57
Therapeutics	894	5.60	563	6.44
Value	411	2.57	344	3.93
Total	15,962	100.00	8,748	100.00

Reference: O: Number of NE Occurrences; OR: Number of NE Occurrences Ratio; DO: Number of Distinct NE Occurrences; DOR: Number of Distinct NE Occurrences Ratio

The entire dataset was annotated by the author of this thesis, a final-year student of the MSc in Biomedical Engineering. After that, to validate the annotation, almost 30% of the dataset (90 texts) was revised by two final-year students of the MSc in Biomedical Engineering, two PhD students in Data Science, one Computer Science Professor working on NLP and NER, and one Physiotherapist. Each of the previous subjects revised 15 texts. Based on the revised subset, we calculated the agreement ratios as the ratio between the number of tokens which were annotated with the same tag as our annotation and the total number of tokens for each NE class. Although there were some tokens annotated with different tags, we did not change dataset labels because it would introduce some bias on the texts as 70% of the dataset was not revised. Therefore, the revision was only to check how well labeled was the dataset. Agreement Ratio (AR) for each NE class, as well as the number of agreed (AT) and of not-agreed tags (NAT) are in table 4.6. It also presents the AR for all the texts, i.e.,

The lowest ARs are for Additional Observations, Characterization and Results. These were also the classes whose original labelling raised more doubts. Additional

Table 4.5: NE classes description for Test Dataset.

NE Class	O	OR (%)	DO	DOR (%)
Anatomical Site	17	2.644	14	2.960
Condition	99	15.397	66	13.953
Characterization	51	7.932	45	9.514
DateTime	130	20.218	102	21.564
Evolution	52	8.087	34	7.188
Genetics	0	0.000	0	0.000
Negation	33	5.132	7	1.480
Additional Observations	47	7.309	34	7.188
Results	19	2.955	17	3.594
Route of Administration	3	0.467	3	0.634
Test	66	10.264	44	9.302
Therapeutics	88	13.686	73	15.433
Value	38	5.910	34	7.188
Total	643	100	473	100

Reference: O: Number of NE Occurrences; OR: Number of NE Occurrences Ratio; DO: Number of Distinct NE Occurrences; DOR: Number of Distinct NE Occurrences Ratio

Observations is a general class which may include other NEs, in case it does not relate to the patient but to their family, e.g. "...diagnóstico de doença neoplástica no marido..." (*...diagnosis of neoplastic disease in her husband...*), or information about the patient that is important but does not suit any other class, e.g. "...abandono do acompanhamento médico..." (*...abandonment of medical assistance...*). Characterization may have tokens from the Condition or Evolution classes, depending on the perspective of the reader, e.g. "possível" (*possible*) in "possível processo vascular" (*possible vascular process*) or "hipótese" (*hypothesis*) in "hipótese de metástase" (*hypothesis of metastasis*) for Condition and "progressivo" (*progressive*) in "declínio cognitivo progressivo" (*progressive cognitive decline*) for Evolution). Depending on their interpretation, Results may also have tokens from Condition (e.g. "nova lesão" (*new injury*) in "...RM-CE que documentou nova lesão..." (*...RM-CE which documents a new injury...*) or "hematoma" in "...TAC-CE que mostrou aumento do hematoma..." (*...TAC-CE which shown an increase of the hematoma...*).

For all NE classes, the agreement is above 90%, except for Characterization. This is high, especially given the number of classes covered and that the used documents

Table 4.6: Agreement Ratios for all NE and Non-Entity classes.

Class	AR (%)	AT	NAT	Total
Anatomical Site	98.01	1,821	37	1,858
Condition	94.16	2,323	144	2,467
Characterization	86.29	428	68	496
DateTime	93.79	1,193	79	1,272
Evolution	97.15	375	11	386
Genetics	100.00	27	0	27
Negation	97.74	259	6	265
Additional Observations	91.11	164	16	180
Results	91.68	1,322	120	1,442
Route of Administration	91.30	21	2	23
Test	96.81	1,273	42	1,315
Therapeutics	95.13	605	31	636
Value	96.78	331	11	342
Out	96.91	8,941	285	9,226

Reference: AT: Number of Agreed Token Tags; NAT: Number of Not-Agreed Token Tags

are not always easy to interpret, due to the high presence of medical terminology. We recall that these numbers apply for only 30% of the dataset. Due to lack of time, the remaining documents were not revised.

4.1.2 Word Embeddings

WEs models are distributional semantic models which are able to convert each word to a vector as explained in section 2.4. For that, they had to be trained with several texts. In order to check which texts were the best for training WEs for clinical NER, we used pre-trained in-domain and out-of-domain models.

For out-of-domain WE model we used a general Portuguese WE model downloaded from the FastText website[6]. It had been trained with billions of tokens from Wikipedia and Common Crawl [88].

Since the out-of-domain WEs were trained with a character window of 5 characters, a total of 27 words and 80 lemmas in our dataset do not have an embedding vector

[6] https://fasttext.cc/docs/en/crawl-vectors.html

in this model, e.g. “IgEV”, “DYSF”. For them, we assigned the WE of the word ’UNK’, given that this word means unknown and it is not Portuguese nor introducing much noise to the embedding datasets. This strategy was followed because simply putting out these words could influence the labelling of the network, as the classification of each word depends on the classification of the others around.

In-domain WE models were trained with 3,377 clinical texts collected from Sinapse journal, comprising 686,762 tokens all together. This journal was chosen once again with the purpose of training the WE models with texts similar to the training/validation ones.

After collecting the texts, we used the FastText algorithm, available in the Gensim library [89], for training the model. We chose this algorithm because it learns not only the context of each word, but also their morphology, since each word is learnt by summing its character n-gram embeddings. This is useful in clinical field as there are words, e.g diseases and therapies names, that share the same prefixes or suffixes which could make their labelling easier. Since the WE model trained using FastText algorithm learns each word by summing each character n-gram, it is able to represent out-of-vocabulary words, which would not be possible using the word2vec algorithm.

For training the FastText model, the following parameters were used: 300-dimension vectors, skip-gram with negative sampling, minimum count of 5 words, minimum char-gram length of 1, and default settings for the remaining hyperparameters. Furthermore, as we had not a large dataset to train the WE model, we used skip-gram algorithm. The number of dimensions (300) and minimum word count (5) were the same as in the out-of-domain WE model. Minimum char-grams length (1) was used for training the model with all the characters, thus enabling to recognize all the unknown words. Finally, during the training of the model, all the words in the dataset starting with an uppercase character were converted to lowercase, since they represent the same word but in the beginning of a sentence. However, if the word contained more than one capital letter it was not converted to lowercase. After preprocessing, only 7,312 out of 26,686 distinct tokens appear more than 5 times in the dataset which means that the word embedding model did not learn all the context of the words present in the dataset since the minimum word count is 5. However, it is able to map the embeddings for the out-of-vocabulary words because it also learnt the morphology of the words.

4.2 Methods

After the preprocessing step, we had to extract the best features from each token, i.e., words and punctuation. Sections 4.2.1 and 4.2.2 present the features extracted from each token and how the best features were chosen. Section 4.2.4 presents a baseline dictionary-based model made just to compare with the complex models. Furthermore, sections 4.2.3 and 4.2.5 show how we proceeded to build our models from the architecture until their grid search. Although the state of the art results are generally achieved with deep learning based models, the CRF allowed us to check which are the best features for sequence labelling in the clinical field because it is a white box model. Therefore, we can see it as a baseline model.

4.2.1 Feature Extraction

For extracting features from the tokens, we considered a 5-token context window (current token, two previous and two following ones) for which the following *baseline* features [32, 72, 66] were extracted:

- Orthographic and Morphological:
    - Token is a punctuation sign, e.g. "." or "/" (Possible values: True or False);
    - Token only has ASCII characters, e.g. "vascular" (Possible values: True or False);
    - Token only has lowercase characters, e.g. "lesão" (*injury*) (Possible values: True or False);
    - Token only has uppercase characters, e.g. "ECG" (Possible values: True or False);
    - Token only has alphabetic characters, e.g. "metástase" (*metastasis*) (Possible values: True or False);
    - Token is numeric, e.g. "7" (Possible values: True or False);
    - Token is alphanumeric, e.g. "q10" (Possible values: True or False);
    - Token starts with an uppercase character, e.g. "Dasatinib" (Possible values: True or False);

  - Token ends in “a”, e.g. “coenzima” (*coenzyme*) (Possible values: True or False). This feature represents the majority of regular Portuguese feminine nouns;
  - Token ends in “s”, e.g. “áreas” (*areas*) (Possible values: True or False). This feature represents the majority of regular Portuguese plural nouns;
  - Token shape where the uppercase characters are converted to “A”, the lowercase to “a”, the numbers to “#” and the punctuations to “-”, e.g. “AngioTC” is converted in “AaaaaAA” and “q10” is converted in “a##”;
  - Token length, e.g. “vascular” has 8 characters;
  - Token prefixes and suffixes. A 5-character window was used for both affixes, e.g. “vascular” prefixes are “v”, “va”, “vas”, “vasc” and “vascu” and the suffixes are “r”, “ar”, “lar”, “ular” and “cular”.
- Linguistic: token, POS tag and lemma.

Besides those features, we added features based on the in-domain WE model, since it got the best results as shown in section 5.2. Distributional features based on this model were: the ten most similar words and the IOB tag, given by the nearest mean and median prototype vectors. The most similar words were those that maximized the cosine with the token word vector.

Prototype vectors were calculated using the WEs of the tokens of the dataset and their labels. First, we separated all the tokens by their IOB tags. Then, as there are some tokens which could belong to more than one IOB tag we used Normalized Pointwise Mutual Information (NPMI) [90] (equation 3.1) to select the most appropriated tag for each token. Afterwards, we convert all the tokens in their WE vector and computed the mean and the median prototype vectors for each IOB tag cluster. Finally, to get the features, we computed the cosine between the token WE vector and each of the IOB tag cluster prototype vectors, and selected the IOB tag of the nearest mean and median prototype vectors.

4.2.2 Feature Selection

As presented in section 2.3, the CRF classifier gives one weight to each of the feature values during training. In other words, for the same feature there are different weights for different values (e.g. the word “Epilepsia” (*Epilepsy*) has a higher weight than “EEG” for the IOB tag “B-C”). We use those weights for finding which were the most relevant features. First, we convert each weight to its absolute value, because, depending on its meaning, weights could be positive or negative. Positive weights correspond to positive association between the feature and the proposed class while negative weights have the opposite meaning. We summed all those values for each feature and selected the feature with the maximum sum (max_{sum}). Then, we selected the features with a sum of weights equal or above a certain threshold, computed by multiplying a certain percent with the max_{sum}. For getting the best threshold, we tested different values from 0% to 100% with 5% steps. We present this as well as the best threshold value in section 5.1.2.

4.2.3 Conditional Random Fields

For training the sequence labelling classifier, we used the sklearn-crfsuite package[7]. In our case, the weights are learned during the training phase, using a gradient descent using the Limited-Memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) [91]. As this training algorithm uses L1 and L2 regularizations, we have to search the ones that get the best performance. For that, we made a grid search which tested all the combinations of L1 and L2 coefficients from 2^{-5} to 2^{5}, with an exponential step of 1 that is presented in the section 5.1.1.

Furthermore, the number of maximum iterations was 100 and during the training phase the model does not learn all possible transitions because we did not want to introduce transitions beyond the ones in the training dataset e.g. the transition from B-C (Beginning Condition NE) to I-AS (Inside Anatomical Site NE) was not learnt by the model.

4.2.4 Dictionary-Based Model

We build a simple model for comparison with the others and, hopefully, motivate the necessity of more complex models. It is based on a dictionary build with training dataset NE occurrences. Figure 4.1 shows an summary of how the model works.

During training phase, the model collects all the NEs occurrences from the training dataset as well as their number of repetitions. Figure 4.2 shows how the dictionary is built.

[7] `https://sklearn-crfsuite.readthedocs.io/en/latest/index.html`

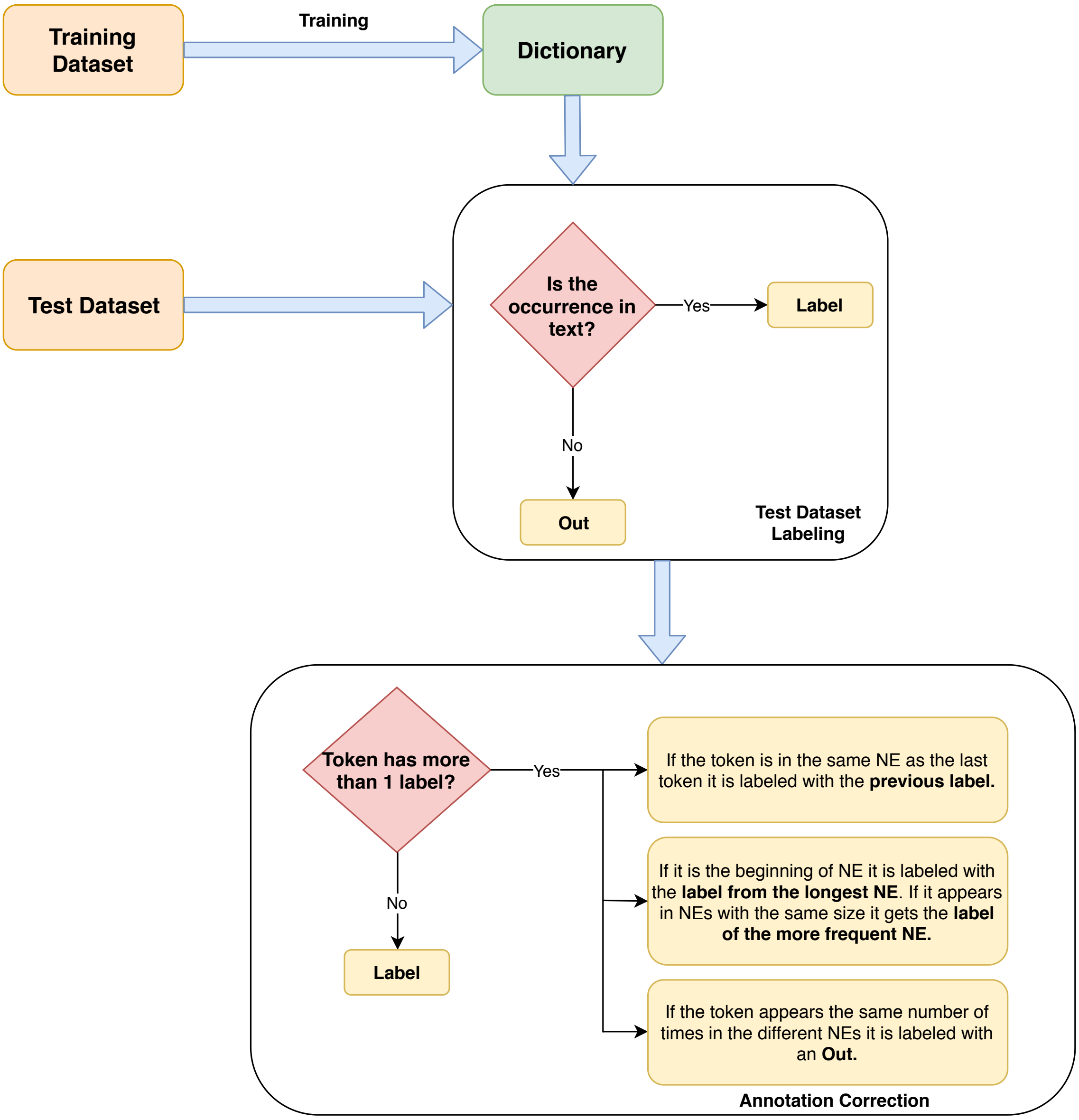

Training Dataset
Training
Dictionary
Test Dataset
Is the occurrence in text?
Yes
Label
No
Out
Test Dataset Labeling
Token has more than 1 label?
Yes
If the token is in the same NE as the last token it is labeled with the **previous label.**
If it is the beginning of NE it is labeled with the **label from the longest NE**. If it appears in NEs with the same size it gets the **label of the more frequent NE.**
If the token appears the same number of times in the different NEs it is labeled with an **Out.**
No
Label
Annotation Correction

Figure 4.1: Dictionary-based Model Architecture.

To note that the number of each NE occurrences were set in order to explain the following examples. They were not set according to the real dataset.

During test phase, this model checks which occurrences of the dictionary are in the texts. If an occurrence is in the text, all the tokens which belong to it are annotated with the respective label. Afterwards since there could be more than one label for each token, we have to create some rules to make each token has just one label.

As shown in figure 4.1 there are four different ways to tag a token if it was labelled with more than one NE class in the previous phase:

- If the token is in the same occurrence of the previous one it is labelled with

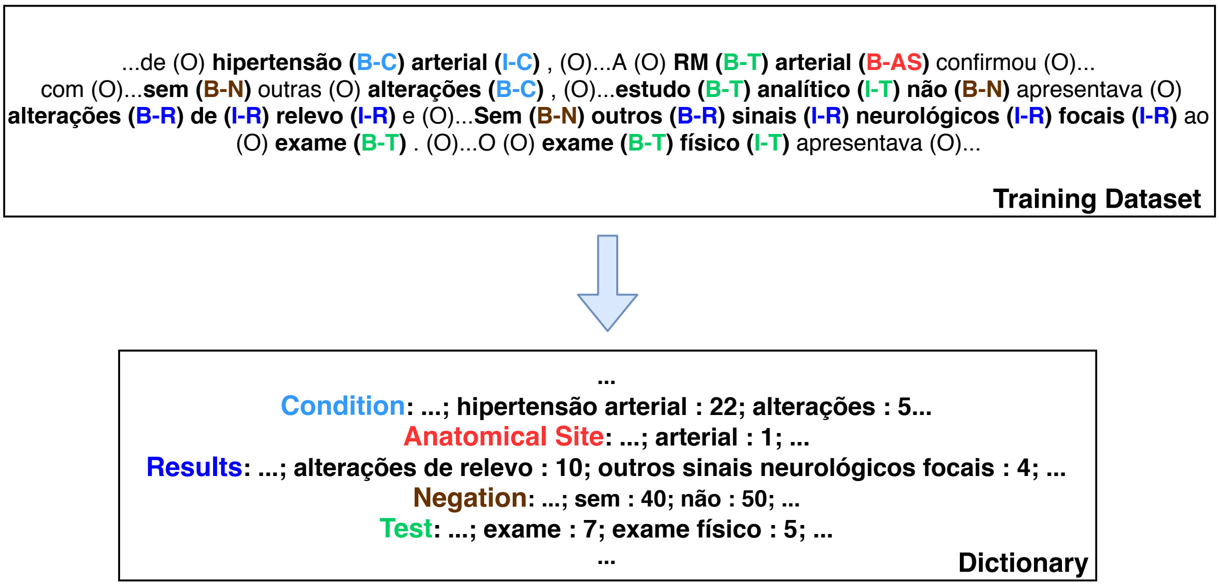

...de (O) hipertensão (B-C) arterial (I-C) , (O)...A (O) RM (B-T) arterial (B-AS) confirmou (O)...
com (O)...sem (B-N) outras (O) alterações (B-C) , (O)...estudo (B-T) analítico (I-T) não (B-N) apresentava (O)
alterações (B-R) de (I-R) relevo (I-R) e (O)...Sem (B-N) outros (B-R) sinais (I-R) neurológicos (I-R) focais (I-R) ao
(O) exame (B-T) . (O)...O (O) exame (B-T) físico (I-T) apresentava (O)...
Training Dataset
...
Condition: ...; hipertensão arterial : 22; alterações : 5...
Anatomical Site: ...; arterial : 1; ...
Results: ...; alterações de relevo : 10; outros sinais neurológicos focais : 4; ...
Negation: ...; sem : 40; não : 50; ...
Test: ...; exame : 7; exame físico : 5; ...
...
Dictionary

Figure 4.2: Example of how the Dictionary Model is trained.

the same NE class, e.g. using the dictionary of figure 4.2, "arterial" in "...tipo 2, hipertensão arterial e..." (*type 2, arterial hypertension and...*) could be labeled as Condition or Anatomical Site. However, as the previous word ("hipertensão") was already labelled as Condition, "arterial" will also be labeled as Condition.

- In case of a beginning of an occurrence, i.e., if the token is in the beginning of a document or if the previous token has the label "Out", the model chooses the label of the longest occurrence, e.g. once again using the dictionary of figure 4.2, "alterações" in "...sem alterações de relevo..." (*...without relevant changes...*) could be labelled as Results or Condition. However, as the NE occurrence of the Results class is longer than the NE occurrence of the Condition class it is labelled as Results. Also this rule works for tokens which appear in different occurrences from the same label, e.g. "exame" (*test*) in "...o exame físico..." (*...the physical test*) could belong to "exame" and "exame físico". Although both belong to Test the model selects the second one because it is longer.
- If there are two or more labels with occurrences of the same size the model selects the label of the occurrence where the token appears more frequently.
- If even after these rules there is a tie between two or more labels the token is labelled as Out and the model continues for the following one.

This model should not be taken as a NER model proposal but rather as a baseline.

4.2.5 Deep Learning Architectures

Given the current trend on NER and its state of the art results, we adopted two architectures based on BiLSTM-CRF neural network as our models for this purpose. The first one is presented in figure 4.3 while the second one, the residual learning model, is based in figure 3.1 but with three stacked Bidirectional Long Short-Term Memory (BiLSTM) layers instead of the two shown in the figure and one CRF layer on the top. Despite their different architectures, both share the same input data, i.e., the vectors presented in figure 4.3.

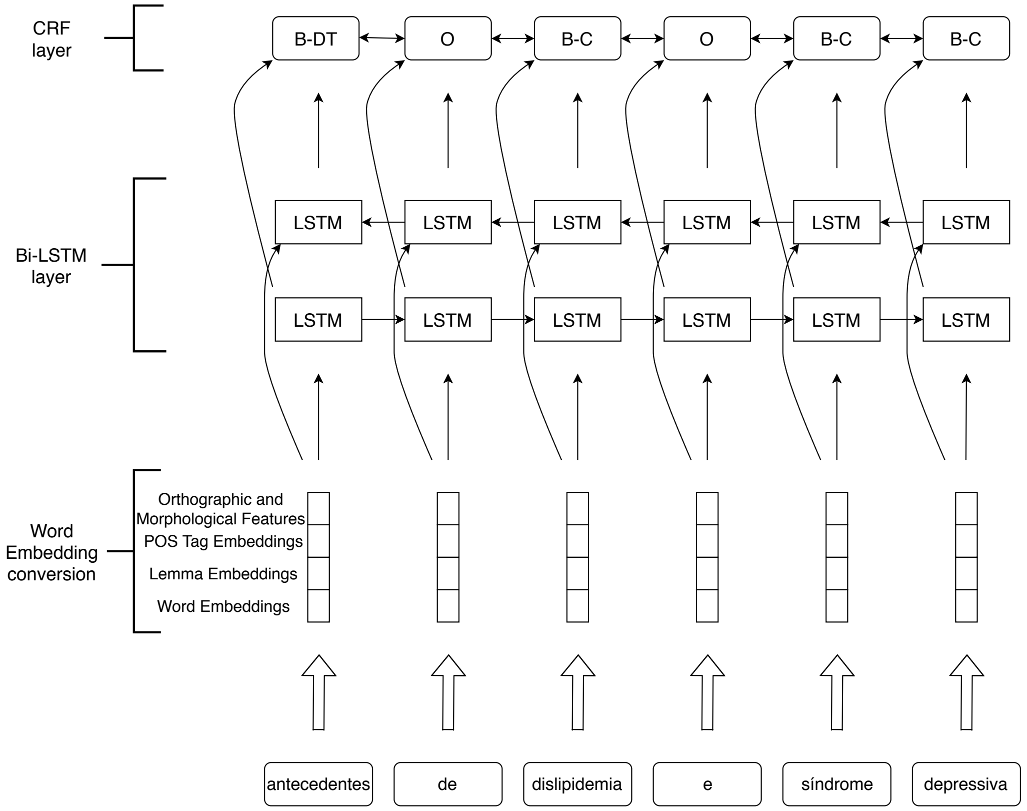


Figure 4.3: BiLSTM-CRF Neural Network Architecture on the sentence: "antecedentes de dislipidemia e síndrome depressiva" (*history of dyslipidemia and depressive syndrome*).

The WE step was where all the tokens were converted to their embedding vectors. Lemmas were also converted to their WE vectors and concatenated to the previous vectors. As POS tags were strings, they had to suffer a transformation in order to be accepted by the neural network. First, we made a list with all the POS tag options and then we transformed each string to an one-hot encoding vector. Afterwards, we concatenated this vector to the previous ones. Finally, all the orthographic and morphological features presented in section 4.2.1 except length, word shape, prefixes and suffixes were also concatenated to the input vector. Although the prefixes and suffixes have been important to the CRF model (section 5.1.2) we did not use them because this information should also be covered by the FastText WEs.

Afterwards, in both models, the embedding vectors were inserted in the BiLSTM layer with one backward layer and one forward layer. The former enables the network to preserve the information from the past to the future, since it analyses the information from the left to the right. The forward layer enables the network to do the inverse of the backward. Together, these types of Long Short-Term Memory (LSTM) improve the prediction of the network, which, this way, understands better the context of each token. After that, in the residual learning model, the second layer takes as input the concatenation of the hidden vector of the first BiLSTM layer with the input vector and the same for the third layer as shown in figure 4.4.

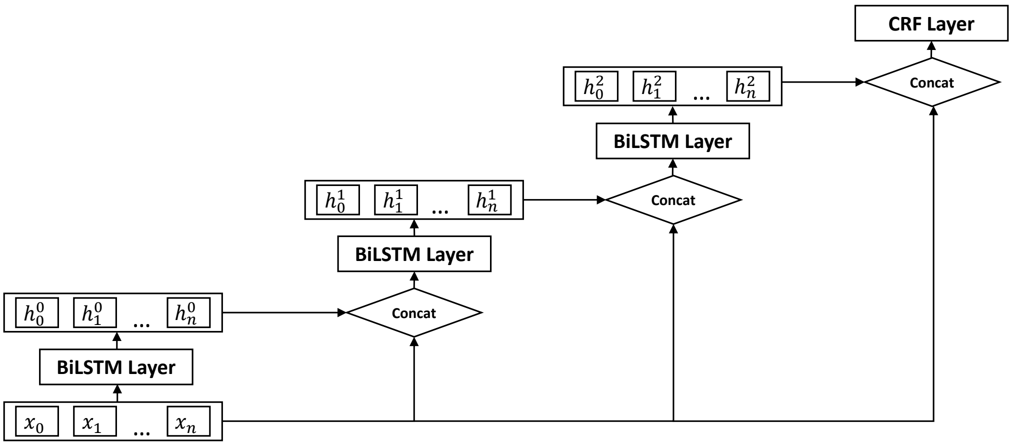


Figure 4.4: Residual Learning Model Architecture. The x vectors are the token embeddings and the h vectors are the hidden vectors of each layer.

Finally, the hidden vector of the BiLSTM layer (the third BiLSTM layer in the residual learning model) was inserted in the CRF layer, which enables the network to consider the neighbour tags. In other words, it allows the network to create tag relations, e.g., if a token is tagged with a beginning of an NE, the following token is probably the continuation of such NE. This layer is also responsible for not allowing a token to be tagged with an in-NE tag without this NE being started previously.

We used Adam optimization function [92], a function that adapts the learning rate according to network parameters using the first and second moments of gradient, i.e., the mean and the variance. According to the authors it combines the advantages of two other optimizer functions AdaGrad and RMSProp, working well with sparse gradients and online and non-stationary settings. It was used with an initial learning rate of 0.001. For finding the best number of hidden units and the best dropout percentage, we made a grid search with 50 training epochs, presented in section 5.1.3. As the dataset had a low number of instances, we used a small set of values for the grid search of the number of hidden units [2^3, 2^7] varied in powers of two. Keeping the network with a low number of parameters prevents overfitting of the network to the data [93]. Furthermore, we used an interval of dropout percentage values from 10% to 50% with a 10% step. This hyperparameter allows the network to prevent both overfitting and under-learning [94].

First, an independent grid search was run for each WE model for BiLSTM-CRF model, because they had been trained with different types of texts. After that, as the in-domain WE model got better results (table 5.5) we performed grid search for residual learning model only using in-domain WEs. Given that the search for the best hyperparameters for each layer would spend much computational time we used the same hyperpameters for all the layers. The range of values were the same as for the BiLSTM-CRF model, i.e, [2^3, 2^7] varied in powers of two for hidden units and [10%, 50%] with a 10% step for dropout percentage.

5

Results and Discussion

This chapter presents the main results of this research, starting with the results of the grid search for each model as well as the search for the best threshold for feature selection. Then, we compare both Word Embeddings (WEs) using the Bidirectional Long Short-Term Memory (BiLSTM)-Conditional Random Fields (CRF) model. Afterwards, we describe the results for each model using not only the Prediction, Recall and F1-Score but also making a statistical comparison between them. Finally, we discuss the results for the independent test set.

5.1 Models Optimization

This section describes the hyperparameter study for each machine learning model. It starts by describing the grid search for the CRF model (section 5.1.1). Then, section 5.1.2 describes the search for the best threshold for feature selection and also presents which features lead to a better performance. The section ends with a description of the grid search results for the deep learning models (section 5.1.3).

5.1.1 Conditional Random Fields Hyperparameters Search

As stated in section 4.2.3, to train a CRF classifier, first we had to find the best parameters for L1 and L2 regularization for the L-BFGS method. Figure 5.1 presents the relaxed micro F1-Score for each pair of parameters, using 10-fold Cross Validation (CV). It shows that the best performances were achieved with the lower pair of values tested. We used 2^{-5} and 1 for L1 and L2 regularization coefficients, respectively, since the classifier trained with this pair of parameters achieved the best performance. These values are consistent with the literature since low L1 and L2 regularization coefficients mean that the more important features take higher weights than the others, while with high regularization coefficients all the features take similar weights, meaning that none of them has discriminative power.

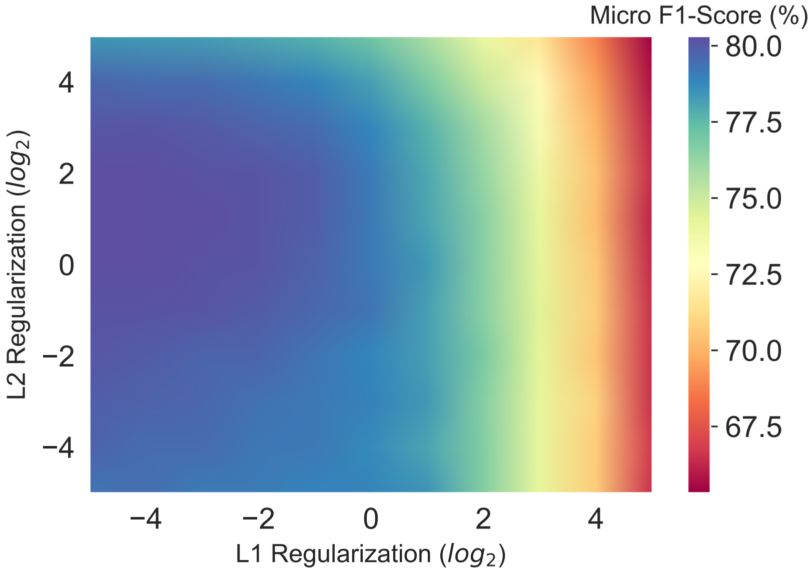


Figure 5.1: Grid Search for finding the best L1 and L2 regularization parameters. The values between each step were calculated using interpolation in order to make the image smoother.

Table 5.1 presents all the parameters used for training the CRF classifier. As stated in section 4.2.3, we did not use all possible transitions because we did not want the model to consider forbidden transitions e.g. transition from B-AS (Beginning Anatomical Site Named Entity (NE)) to I-C (Inside Condition NE). However, we used all possible states, i.e., all possible feature function combinations that are not related with any classes in the training dataset. These are called negative feature functions and could improve model performance by learning associations between features and labels that do not appear on the training dataset.

Table 5.1: Summary of CRF hyperparameters.

Hyperparameters	**Values**
Training Algorithm	Limited-Memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS)
L1 Coefficient	2^{-5}
L2 Coefficient	1
Maximum Iterations	100
All Possible Transitions	False
All Possible States	True

5.1.2 Feature Selection Threshold Search for Conditional Random Fields

After finding the best parameters for the CRF, we searched for the best threshold for the feature selection method using a 10-Fold CV, i.e., we obtained the performance for each threshold performance making 10-Fold CV and then we selected the percentage with the best average performance. Figure 5.2 shows the relaxed micro F1-score for each percentage threshold tested. It shows that the best performances were achieved for the thresholds between 0% and 20%, with 0% and 10% achieving the maximum micro F1-score. The abrupt fall at 90% is due to the result of removing important features, such as similar words and affixes from the current and surrounding tokens. Using a percentage threshold of 10% decreases the number of features from 185 to 109, also reducing the computational processing time but maintaining the same performance.

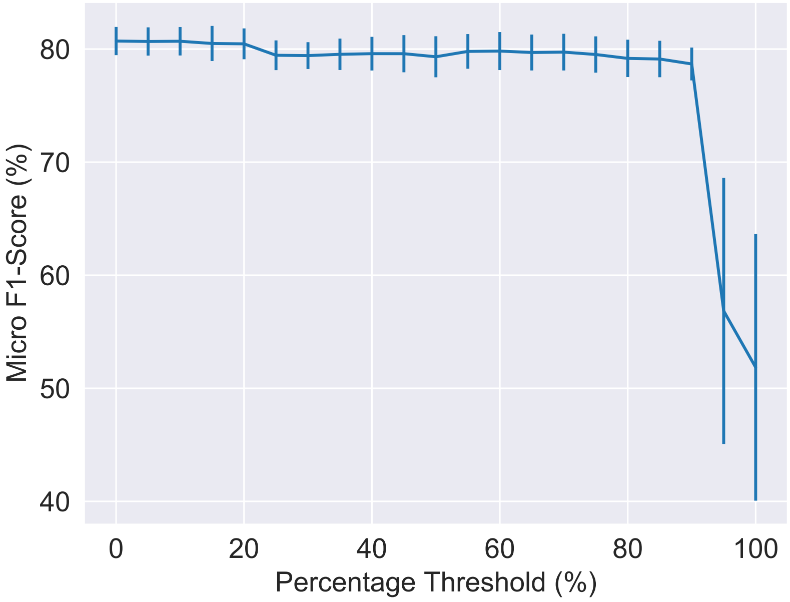


Figure 5.2: Grid Search for finding the best percentage threshold for feature selection method. The results were obtained by making the average and standard deviation of the performances for each 10-Fold CV.

The best features for all the folds using a threshold of 10% were: all the similar words; all the lemmas; only the nearest mean and median prototype vector indexes of the current token; all the prefixes with more than one character except the prefixes of the current token which also includes the prefix with just one character, all the suffixes with more than one character, all tokens, all the token lengths and the current token shape.

Among others, similar words are very useful for finding the real meaning of abbreviations, because their extended versions are typically in the top of their nearest words. For instance the most similar words for “EEG” are “Encefalografia” (*Encephalography*) and “Encefalograma” (*Encephalogram*). Not to mention that it is expected that nearest words belong to the same class. Tokens are essential because they are what is actually mentioned. Also, lemmas are significant as they allow to relate each Inside-Outside-Beginning (IOB) tag with the dictionary form of the word. This relation enables to classify with the same IOB tag even if the words are in inflected differently, e.g. “tromboses” (*thromboses*) and “trombose” (*thrombosis*). However, these last two features may lead to overfitting since they behave like a dictionary.

The features from the prototype vectors support the classification, because each vector carries information of each IOB tag. It was already expected that the prefix and suffix features would be very relevant, because medical documents typically use many words with them. For example, the prefix “dis” means difficulty, “exo” means out, and “meta” alteration; the suffix “ase” means enzyme, “ismo” means disease and “oma” tumor.

It was expected that Part-of-Speech (POS) tags were important for each token classification, since they represent the grammatical function of each one and thus support sequential classification, i.e., some POS, such as nouns or adjectives, are prone to be the beginning of an entity. The feature word shape is also significant because it carries almost all the morphological information for classifying entities as Value and DateTime. For instance, the token “18” is converted to “##” and the token “2/3” is converted to “#-#”. Token lengths could be meaningful for classifying tokens since there are some classes such as Value and Route of Administration that are mainly constituted by small tokens and others like Therapeutics, Anatomical Site or Condition that have several long tokens.

5.1.3 Deep Learning Architectures Optimization

After selecting the best hyperparameters for CRF model and the best threshold for feature selection, we proceeded to find the hyperparameters for our deep learning models which lead to the greatest performance using 10-Fold CV. Figure 5.3 provides the grid search results for the number of Long Short-Term Memory (LSTM) units and dropout percentage for the BiLSTM-CRF model that uses in-domain WEs.

As in the previous figure 5.1 the values between each step were calculated using interpolation in order to make the image smoother. The area with 64 LSTM units contains the highest results. The best pair found in this area is 64 LSTM units and 50% dropout percentage.

As the previous figure, figure 5.4 also has one optimal area. However, in this case the area is found for 32 LSTM units rather than 64. The best number of LSTM units and dropout percentage are 32 and 50%, respectively. Finally, figure 5.5 presents the performances when using different parameters for the residual learning model.

Unlike the previous two cases, this model seems to have two optimal areas. However, the highest result was found for 128 LSTM units and 40% dropout percentage. Tables 5.2 and 5.3 summarize the hyperparameters used on both BiLSTM-CRF models. Table 5.4 contains the hyperparameters used on the residual learning model.

Table 5.2: Summary of the hyperparameters used on BiLSTM-CRF model with in-domain WEs.

In-Domain WEs	
Hyperparameters	**Values**
LSTM Units	64
Dropout Percentage	50%
Optimizer Function	Adam
Learning Rate	0.001
Training Epochs	50

Table 5.3: Summary of the hyperparameters used on BiLSTM-CRF model with out-of-domain WEs.

Out-of-Domain WEs	
Hyperparameters	**Values**
LSTM Units	32
Dropout Percentage	50%
Optimizer Function	Adam
Learning Rate	0.001
Training Epochs	50

Table 5.4: Summary of the hyperparameters used on Residual Learning Model.

Hyperparameters	**Values**
LSTM Units	128 each layer
Dropout Percentage	40%
Optimizer Function	Adam
Learning Rate	0.001
Training Epochs	50

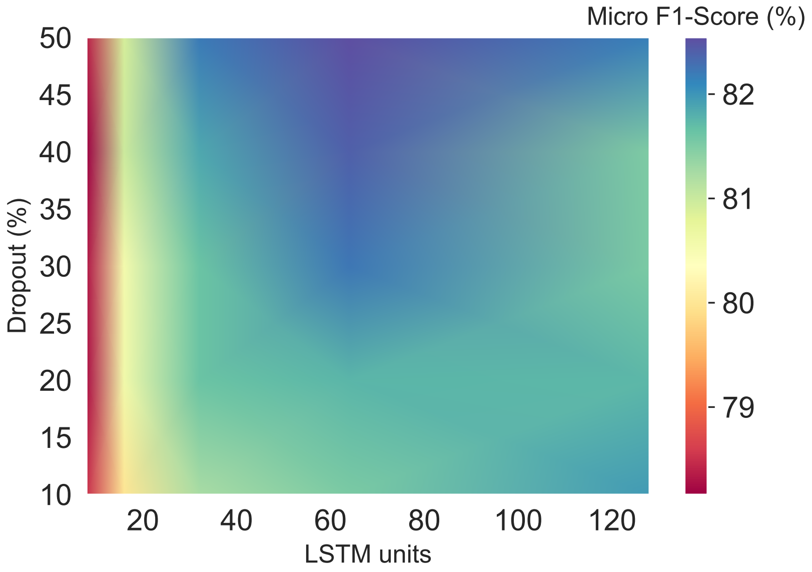


Figure 5.3: Grid Search for finding the best number of LSTM units and dropout percentage for BiLSTM-CRF model that uses in-domain WEs. The values between each step were calculated using interpolation in order to make the image smoother.

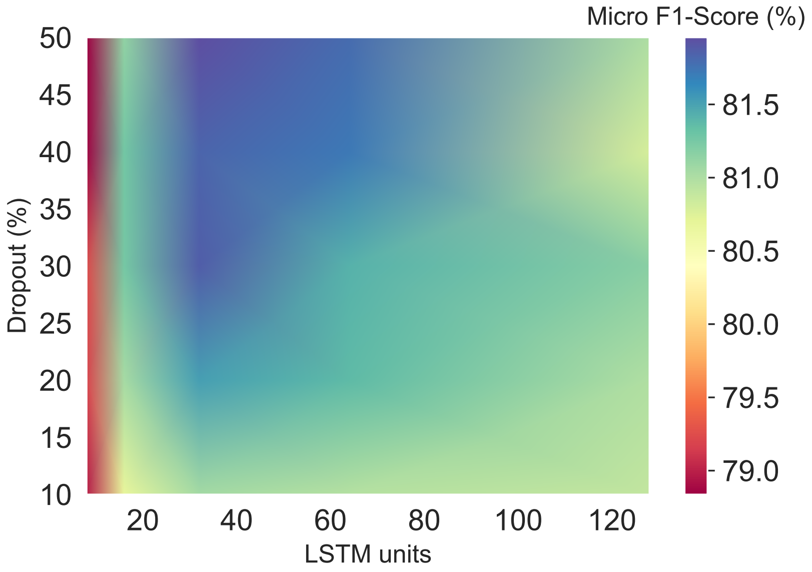


Figure 5.4: Grid Search for finding the best number of LSTM units and dropout percentage for BiLSTM-CRF Model that uses out-of-domain WEs. The values between each step were calculated using interpolation in order to make the image smoother.

The results corroborate that dropout regularization helps avoiding overfitting, since the best results were obtained for high dropout percentage.

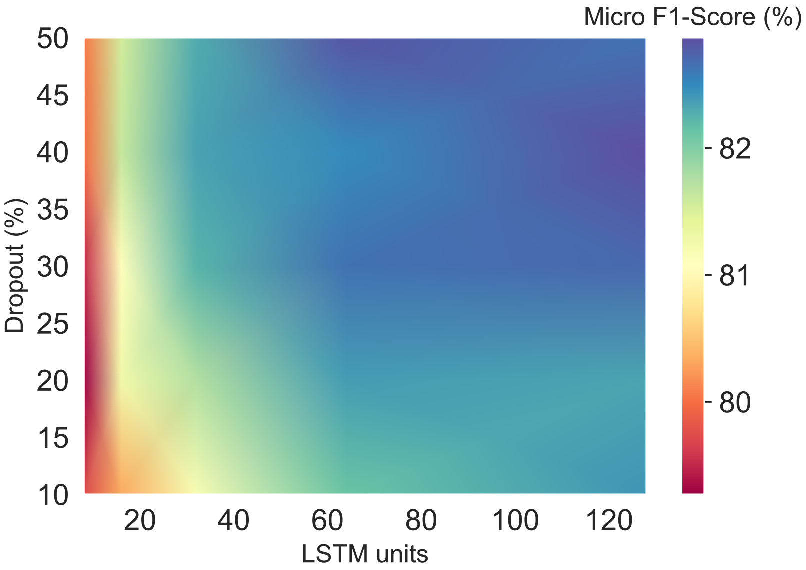


Figure 5.5: Grid Search for finding the best number of LSTM units and dropout percentage for the Residual Learning Model. The values between each step were calculated using interpolation in order to make the image smoother.

5.2 Word Embeddings

After finding the optimal hyperparameters for each model, we had to check what WE model achieved the highest performance. We made ten times random 10-Fold CV because we wanted to have several performances to enable better statistical comparisons.

Besides looking at recall and precision, we focus our discussion on the F1-score. Table 5.5 shows relaxed and strict results for both WE models. The results as well as the next ones are presented in the following format *metric average±metric standard deviation*.

Results show that the in-domain WE model performs better than the out-of-domain. An important reason for this is that the out-of-domain model was not trained with unigrams, leading to the representation of some tokens with the “UNK” vector (27 tokens (0.05% of the training dataset) and 80 lemmas (0.14% of the training dataset)), instead of the original token, thus introducing bias. A second reason is that the out-of-domain model was not trained specifically for the clinical domain. Although trained in a much larger collection of text, the out-of-domain model fails to learn many clinical relations between different diseases or diagnostic tests, as the in-domain model does. Table 5.8 shows examples that confirm this fact, e.g. in the

Table 5.5: Validation results for both WEs.

		Recall		**Precision**		**F1-Score**	
WE	**Average**	**Relaxed**	**Strict**	**Relaxed**	**Strict**	**Relaxed**	**Strict**
In-Domain	**Micro**	82.23±1.57	74.73±1.54	83.01±1.54	75.44±1.69	82.62±1.43	75.08±1.51
Out-of-Domain		82.11±1.42	73.64±1.73	82.75±1.56	75.55±1.47	82.43±1.40	74.58±1.52
In-Domain	**Macro**	78.85±2.47	73.14±2.58	81.21±2.50	75.54±2.78	79.44±2.12	73.84±2.42
Out-of-Domain		77.48±2.65	70.52±2.72	80.70±3.08	74.57±3.15	78.41±2.52	71.90±2.63
In-Domain	**Weighted**	82.23±1.57	74.73±1.54	83.04±1.49	75.39±1.67	82.45±1.45	74.93±1.51
Out-of-Domain		82.11±1.42	73.64±1.73	82.69±1.57	75.27±1.49	82.20±1.41	74.29±1.54

Table 5.6: Relaxed p-values for both WE Models.

Model	**Statistical Test**	**p-value**
In-Domain WE	Kolmogorov-Smirnov	0.263
Out-of-Domain WE		0.988
In-Domain WE	Shapiro-Wilk	0.014
Out-of-Domain WE		0.586

Table 5.7: Strict p-values for both WE Models.

Model	**Statistical Test**	**p-value**
In-Domain WE	Kolmogorov-Smirnov	0.730
Out-of-Domain WE		0.957
In-Domain WE	Shapiro-Wilk	0.472
Out-of-Domain WE		0.985

in-domain model the word "ECG" is related to three other cardiac diagnostic tests, beyond its extended form, while in the out-of-domain model, it is only related to one more word "ecodoppler" beyond its extended form; or the neighbours of "diabetes" in the in-domain model, which include related diseases (e.g., "dislipidemia" (*dyslipidemia*) and arterial hypertension ("HTA")), while, in the out-of-domain model, the neighbours of the same word are words that contain it (e.g., "pré-diabetes" and "diabetes.O"). Furthermore, in the out-of-domain model, several words are not related with the clinical domain, as "hemiparasita" (*hemiparasite*) in the "hemiparésia" (*hemiparesis*) example, or words are not related with anything understandable, as in the "poliangeíte" (*polyangiitis*) example.

After these two comparisons, we performed statistical tests to check if the difference was relevant. First, we performed Kolmogorov-Smirnov and Shapiro-Wilk tests to

Table 5.8: Top-5 Nearest Neighbours for both WE Models.

WE	**Word**	**Top-5 Nearest Neighbours**
In-Domain	ECG	ECG-Holter; electrocardiograma; ecodoppler; ecocardiograma; ecocardiogramas
Out-of-Domain	ECG	eletrocardiograma; Electrocardiograma; electrocardiograma; ecocardiograma; Ecocardiograma
In-Domain	diabetes	mellitus; dislipidemia; dislipidémia; HTA; diabética
Out-of-Domain	diabetes	diabete; pré-diabetes; Diabetes; Pré-diabetes; diabetes.O
In-Domain	paramnésia	amnésia; amnésico; mnésico; mnésica; desorientação
Out-of-Domain	paramnésia	paramécia; param3; paranóia.; alucinatória; articulatória
In-Domain	polineuropatia	neuropatia; mononeuropatia; axonal; sensitivo-motora; miopatia
Out-of-Domain	polineuropatia	Polineuropatia; polineuropatias; mononeuropatia; polineurite; neuropatia
In-Domain	poliangeíte	ganglonopatia; citopatia; mielopatia; linfoproliferativa; granulomatosa
Out-of-Domain	poliangeíte	CH12CH14CH15CH18CH26CH30CH4DH5DH6DH8DH9DH10DH12DH15DH20DH30DH; estômagoCarbosymagDulcolaxGavisconImodiumIpraaloxLansoylLubentylMaaloxMicrolaxRennieSmectaSpasfon; XIII787980818283848586878889909192Colóquio; AnguloSimulacrosVeículosABCIABSCABTDABTMBRTPBRTSBSRPBSRSLTRGVAMEVAPAVCOCVCOTVEVECIVETAVFCIVGEOVLCIVOPEVPVPMEVPMTVRCIVSAEVSAMVSATVTGCVTPGVTPTVTTFVTTRVTTUVUCIA1; biológicoCaméfitoLigações
In-Domain	hemiparésia	hemiparesia; hemiplegia; hemianopsia; hemianópsia; biparésia
Out-of-Domain	hemiparésia	hemiparéticos; hemiparesia; hemiparasita; hemiplegia; hemiparasitas
In-Domain	artralgias	poliartralgias; algias; mialgias; cervicalgias; lombalgias
Out-of-Domain	artralgias	Artralgias; artralgia; mialgias; Mialgias; Nevralgias

check if they follow a normal distribution in order to choose the comparison tests. If they follow a normal distribution, the comparison test will be done using parametric tests, otherwise non-parametric. As our model performances do not depend on each other, we have to perform independent statistical tests (independent t-test in case of parametric data and Mann-Whitney otherwise).

Considering an α of 0.05 we found that the relaxed performance samples do not follow a normal distribution contrary to the strict ones (tables 5.6 and 5.7). Therefore, we performed Mann-Whitney test to compare the first ones and independent t-test to compare the second ones. We used Scipy[1] package for running all the statistical tests. Figures 5.6 and 5.7 present the boxplots for each comparison as well as the

[1] `www.scipy.org/`

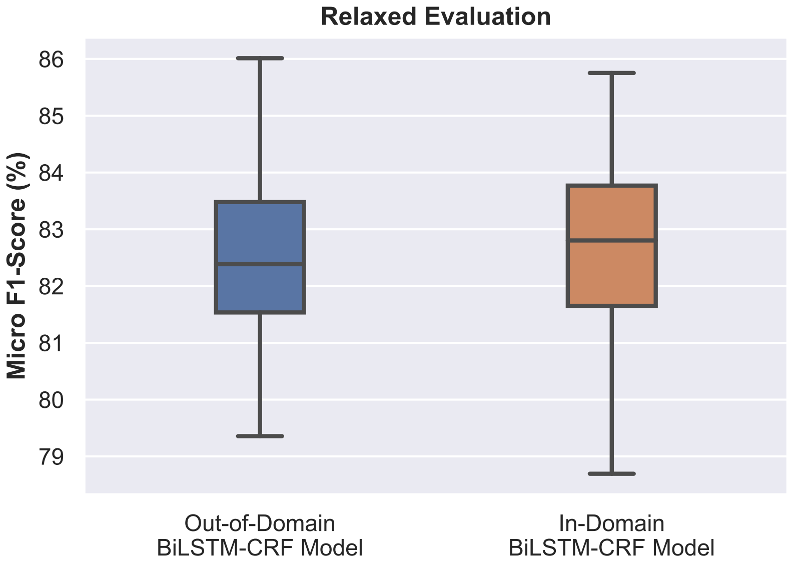


Figure 5.6: In-Domain and Out-of-Domain WEs boxplots using relaxed evaluation performances (p-value = 0.228).

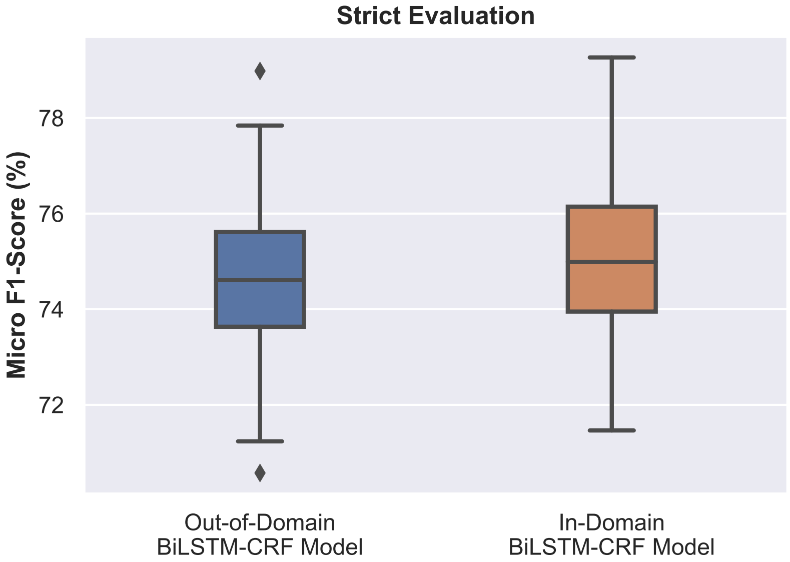


Figure 5.7: In-Domain and Out-of-Domain WEs boxplots using strict evaluation performances (p-value = 0.017).

p-value of such comparison. Again, with an α equals to 0.05 we conclude that, for strict evaluation, the WE models show significant statistical differences, while, for relaxed evaluation, the statistical difference is not significant. However, as the p-value is not high, it is not a strong significance, which means that, possibly, the addition of more samples would lead to significant differences.

Therefore, based on all the results obtained, we may consider that the in-domain WE model performs better than the out-of-domain one.

5.3 Named Entity Recognition Models Validation

Table 5.9 shows the validation results for all the models obtained from ten random 10-Fold CV. It contains three different averages for a better overview of the validation performances.

Table 5.9: Relaxed and Strict validation results of all Models.

		Recall		Precision		F1-Score	
Model	**Average**	**Relaxed**	**Strict**	**Relaxed**	**Strict**	**Relaxed**	**Strict**
Residual Learning	**Micro**	82.64±1.62	75.19±1.73	83.17±1.68	74.79±1.72	82.90±1.56	74.98±1.62
BiLSTM-CRF		82.23±1.57	74.73±1.54	83.01±1.54	75.44±1.69	82.62±1.43	75.08±1.51
CRF		79.58±1.69	71.45±1.72	81.37±1.54	74.79±1.72	80.46±1.57	73.08±1.66
Dictionary-Based		54.09±2.00	51.32±1.77	73.43±1.90	48.17±2.21	62.27±1.65	49.68±1.77
Residual Learning	**Macro**	79.16±2.47	73.69±2.73	81.32±3.07	74.62±3.25	79.64±2.50	73.69±2.73
BiLSTM-CRF		78.85±2.47	73.14±2.58	81.21±2.50	76.85±3.31	79.44±2.12	71.42±2.61
CRF		73.46±2.53	68.18±2.68	82.70±2.92	75.44±1.69	76.55±2.28	75.08±1.51
Dictionary-Based		54.79±2.52	52.23±2.44	70.41±3.37	50.82±3.00	59.68±2.38	50.19±2.33
Residual Learning	**Weighted**	82.64±1.62	75.19±1.73	83.42±1.57	74.98±1.68	82.83±1.53	74.98±1.60
BiLSTM-CRF		82.23±1.57	74.73±1.54	83.04±1.49	75.39±1.67	82.45±1.45	74.93±1.51
CRF		79.58±1.69	71.45±1.72	81.28±1.62	74.67±1.78	79.97±1.65	72.75±1.69
Dictionary-Based		54.09±2.00	51.32±1.77	77.74±1.46	52.32±1.77	62.94±1.71	51.22±1.67

Considering the micro average F1-Score performances, the residual learning model got the highest results over all the models for the relaxed evaluation, i.e, for the token evaluation. However, for the strict performance it got worse results than the BiLSTM-CRF model. We conclude that both models got almost the same performance during validation so that it is not possible to say that one outperformed the other.

Furthermore, these results also confirm the state of the art of Named Entity Recognition (NER), with the deep learning models getting better results than the CRF model. It is known that deep learning models are able to extract valuable features which are hidden to humans. A BiLSTM layer allows to analyse all the inputs keeping information from them, from the beginning until the end of a sequence in both directions. This characteristic could be responsible for the higher validation performances of such models in opposition to the CRF.

Considering the results of the dictionary-based model, it is possible to conclude that the machine learning models perform better because they does not only consider the context to handle ambiguities, but also they can predict NE classes for tokens that were never seen before based on their features.

Table 5.10: Relaxed evaluation p-values for all Models.

Model	Statistical Test	p-value
Residual Learning	Kolmogorov-Smirnov	0.977
BiLSTM-CRF		0.263
CRF		0.483
Residual Learning	Shapiro-Wilk	0.242
BiLSTM-CRF		0.014
CRF		0.060

Table 5.11: Strict evaluation p-values for all Models.

Model	Statistical Test	p-value
Residual Learning	Kolmogorov-Smirnov	0.456
BiLSTM-CRF		0.730
CRF		0.181
Residual Learning	Shapiro-Wilk	0.015
BiLSTM-CRF		0.472
CRF		0.173

In order to check whether the differences in the performances of the models were statistically significant, we perform statistical tests with them. As the dictionary-based model got performances with a high difference from the others, we did not perform statistical tests for it.

Once again, as in section 5.2, we first check if the models performances follow a normal distribution. Tables 5.10 and 5.11 contain the p-values for the three models for both evaluations. As in section 5.2, with an α of 0.05 to evaluate the p-values, we consider that the relaxed evaluation performances of BiLSTM-CRF model do not follow a normal distribution since the p-value of the Shapiro-Wilk test was under the value of the α. Taking into account the strict evaluation performances, we consider that the residual learning model performances do not follow a normal distribution because of the p-value of the Shapiro-Wilk test.

After finding which model performances follow a normal distribution, we made a statistical test to compare the performances of the three models. As in both cases there is at least one model not following a normal distribution, we had to make a non parametric test to compare the three models. First, we made a Kruskal-Wallis test to check if the three models performances came from the same distribution.

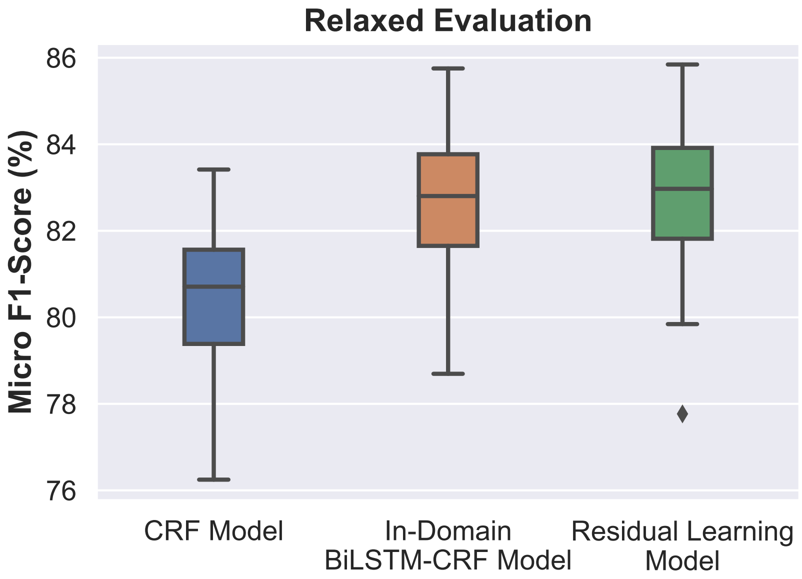


Figure 5.8: All Models boxplots using relaxed evaluation performances (p-value = 1.257×10^{-22}).

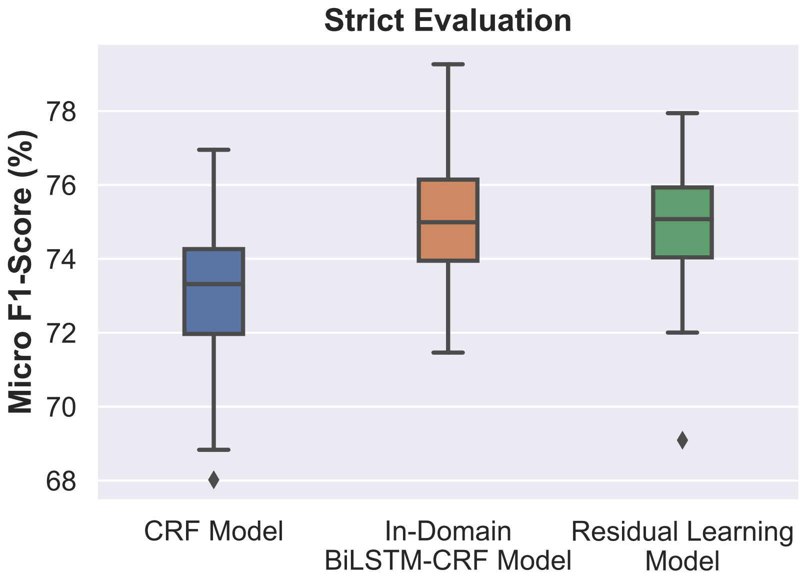
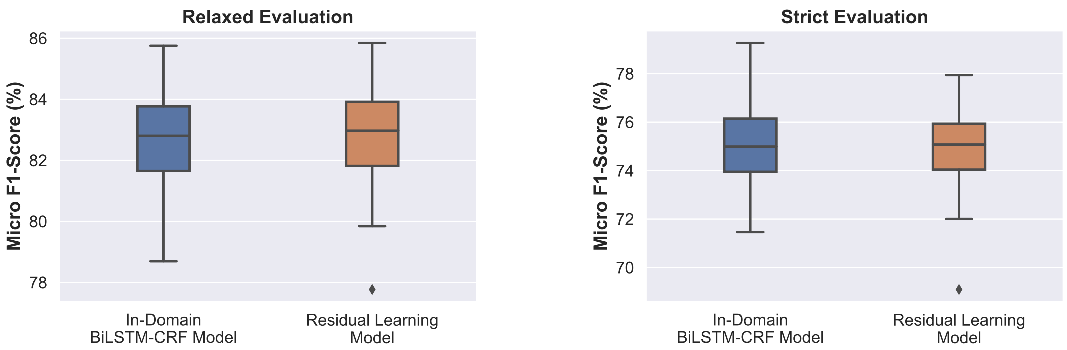


Figure 5.9: All Models boxplots using strict evaluation performances (p-value = 1.112×10^{-16}).

Figures 5.8 and 5.9 show the boxplots for all the model performances. As their p-values were below the value of α, we had to compare the models two by two.

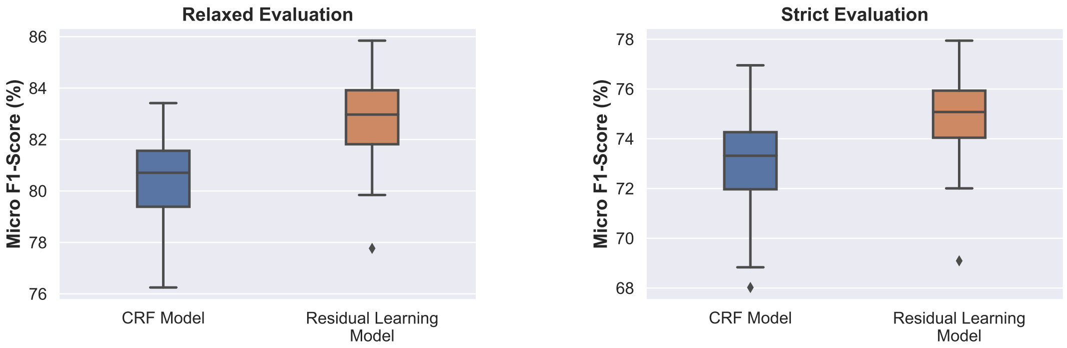
The α value measures the probability of rejecting the null hypothesis when it is true.



(a) Boxplots using strict evaluation performances (p-value = 0.248).

(b) Boxplots using strict evaluation performances (p-value = 0.987).

Figure 5.10: Comparison between residual learning model and BiLSTM-CRF model performances.



(a) Boxplots using strict evaluation performances (p-value = 1.674×10^{-19}).

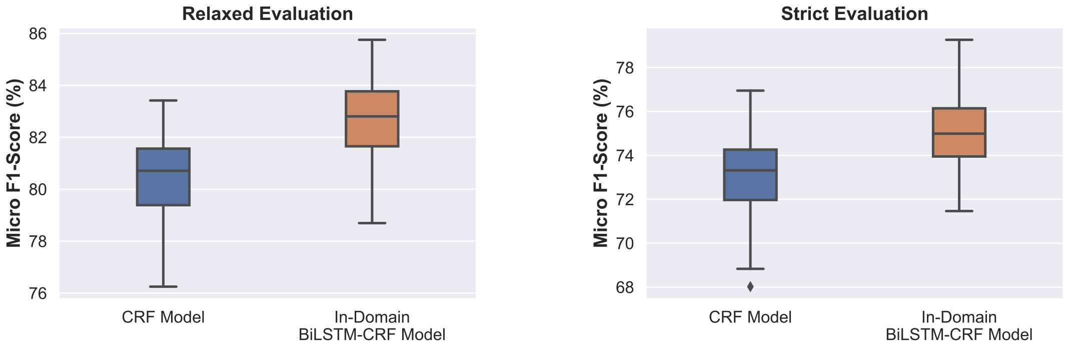
(b) Boxplots using strict evaluation performances (p-value = 9.784×10^{-14}).

Figure 5.11: Comparison between residual learning model and CRF model performances.

It may also be called as type I error. Therefore, the probability of not rejecting the null hypothesis is (1-α) and the probability of rejecting at least one null hypothesis when it is true is calculated by (1-(1-$\alpha)^N$), being the N the number of comparisons that we made.

As we had to make multiple comparisons, if we do not correct the value of the α, the probability of rejecting at least one null hypothesis when it is true will increase. In order to avoid increasing this probability, we had to adjust the value of α using Bonferroni Correction [95]. It consists of dividing the value of α by the number of comparisons that we had to make. After this, our α changed from 0.05 to 0.017.

After getting the results of both deep learning models (Table 5.9), it was expected that the performances of the residual learning and BiLSTM-CRF models did not



(a) Boxplots using strict evaluation performances (p-value = 4.837×10^{-17}).

(b) Boxplots using strict evaluation performances (p-value = 3.988×10^{-16}).

Figure 5.12: Comparison between BiLSTM-CRF model and CRF model performances.

present significant statistical differences, as none of the cases had a p-value below the value of α. However, it could be possible that, if we made a better grid search on the residual learning method, e.g. making the grid search for each layer individually instead of making it globally, better results were obtained.

Once again, both deep learning models got significant statistical differences when compared to the CRF model, which agrees with the current state of the art.

5.4 Model Evaluation on Coimbra Hospital and Universitary Centre (CHUC) Test Dataset

After getting the best hyperparameters and validation performances for each NER model, we tested them on an independent dataset. As explained in section 4.1, these texts were collected directly from CHUC, a different distribution, and it is possible that they have some orthographic errors. Some of them are also written in items, which introduces some bias because the training/validation dataset did not contain text in this form.

Although deep learning models got the best validation results, we decided to test the CRF and the dictionary-based models as well on this test set, in order to check whether the test results are consistent with the validation performances. Table 5.12 and 5.13 presents the performance for each NE class and average performances, respectively, for a more detailed discussion. In order to make the analysis of each NE class simpler, we will mostly consider the results for the residual learning model.

Table 5.12: Results of all models on independent Test Set.

		Recall		Precision		F1-Score	
Model	**NE Class**	**Relaxed**	**Strict**	**Relaxed**	**Strict**	**Relaxed**	**Strict**
Residual Learning	**AS**	100.00	88.24	80.56	71.43	89.23	78.95
BiLSTM-CRF		100.00	88.24	80.56	68.18	89.23	76.92
CRF		86.21	70.59	42.37	40.00	56.82	51.06
Dictionary-Based		68.97	58.82	80.00	55.56	74.04	57.14
Residual Learning	**C**	63.94	64.65	64.25	58.18	64.10	61.24
BiLSTM-CRF		70.19	70.71	59.11	54.26	64.18	61.40
CRF		72.12	61.62	52.63	42.07	60.85	50.00
Dictionary-Based		43.75	41.41	73.39	41.00	54.82	41.21
Residual Learning	**CH**	22.22	19.61	44.90	32.26	29.73	24.39
BiLSTM-CRF		24.24	23.53	42.11	38.71	30.77	29.27
CRF		15.15	21.57	50.00	44.00	23.26	28.95
Dictionary-Based		17.17	11.76	20.00	8.00	18.48	9.52
Residual Learning	**DT**	87.04	65.38	86.77	72.65	86.90	68.83
BiLSTM-CRF		85.80	66.15	84.50	71.07	85.15	68.53
CRF		82.41	48.46	76.95	64.29	79.58	55.26
Dictionary-Based		54.94	46.92	90.82	61.00	68.46	53.04
Residual Learning	**EV**	78.13	73.08	96.15	95.00	86.21	82.61
BiLSTM-CRF		81.25	75.00	82.54	81.25	81.89	78.00
CRF		60.94	51.92	92.86	90.00	73.58	65.85
Dictionary-Based		64.06	57.69	80.39	66.67	71.30	61.86
Residual Learning	**N**	96.97	96.97	86.49	86.49	91.43	91.43
BiLSTM-CRF		96.97	96.97	88.89	88.89	92.75	92.75
CRF		93.94	93.94	91.18	91.18	92.54	92.54
Dictionary-Based		93.94	93.94	88.57	88.57	91.18	91.18
Residual Learning	**OBS**	19.05	10.64	62.50	23.81	29.20	14.71
BiLSTM-CRF		17.14	12.77	64.29	40.00	27.07	19.35
CRF		4.76	6.38	100.00	75.00	9.09	11.76
Dictionary-Based		17.14	10.64	100.00	27.78	29.27	15.38
Residual Learning	**R**	54.55	57.89	58.06	45.83	56.25	51.16
BiLSTM-CRF		63.64	68.42	38.18	44.83	47.73	54.17
CRF		54.55	42.11	19.78	22.22	29.03	29.09
Dictionary-Based		30.30	36.84	66.67	53.85	41.67	43.75
Residual Learning	**RA**	33.33	33.33	100.00	100.00	50.00	50.00
BiLSTM-CRF		33.33	33.33	50.00	50.00	40.00	40.00
CRF		33.33	33.33	100.00	100.00	50.00	50.00
Dictionary-Based		33.33	33.33	100.00	100.00	50.00	50.00
Residual Learning	**T**	86.27	68.18	56.77	46.88	68.48	55.56
BiLSTM-CRF		62.75	54.55	68.82	59.02	65.64	56.69
CRF		50.98	34.85	43.70	33.33	47.06	34.07
Dictionary-Based		30.39	31.82	73.81	52.50	43.06	39.62
Residual Learning	**THER**	79.59	64.77	69.64	60.00	74.29	62.30
BiLSTM-CRF		84.35	67.05	58.49	57.84	69.08	62.11
CRF		69.39	61.36	82.93	80.60	75.56	69.68
Dictionary-Based		36.05	31.82	88.33	53.85	51.21	40.00
Residual Learning	**V**	96.00	84.21	88.89	76.19	92.31	80.00
BiLSTM-CRF		96.00	84.21	88.07	80.00	91.87	82.05
CRF		86.00	63.16	82.69	63.16	84.31	63.16
Dictionary-Based		69.00	65.79	78.41	52.08	73.40	58.14

Reference: CH: Characterization; T: Test; EV: Evolution; G: Genetics; AS: Anatomical Site; N: Negation; OBS: Additional Observations; C: Condition; R: Results; DT: DateTime; THER: Therapeutics; V: Value; RA: Route of Administration

In general, the results for the deep learning methods of table 5.12 follow the agreement ratios presented in table 4.6 since the classes with higher agreement ratio present the best test performances, e.g. Anatomical Site (AS) and Negation (N). This was already expected as the higher agreement ratio the lesser difficulty in labelling with the right class.

The lowest performances were in Additional Observations (OBS) and Characterization (CH) classes. This was already expected for the former, because this class is too general and its labelling consists of tokens that do not belong to any other class (e.g., “restantes irmãos” (*remaining siblings*) and “abandono do acompanhamento médico” (*doctor abandonment*)). As for Characterization, its labelling depends on annotator reading, as some tokens can be Characterizations or Conditions (e.g., “suspeita” (*suspection*) in “suspeita de Arterite de Takayasu” (*suspection of Takayasu's arteritis*) or “hipótese” (*hypothesis*) in “hipótese de AAC” (*hypothesis of AAC*)) , which adds noise to the model. Thus, these two classes are easily labelled by the models as a more specific NE class (e.g. Condition (C) or Evolution (EV)) as explained in section 4.1.

Value (V), Negation, DateTime (DT), Evolution and Anatomical Site show the highest results because they are very specific. Anatomical Site has many words repeated through the dataset that appear in similar contexts like, e.g. “temporal” in “actividade paroxística temporal posterior” *(posterior temporal paroxysmal activity*) and in “córtex temporal superior” *(superior temporal cortex*). It has also different words that appear in similar situations (e.g., “parietal” and “justa-ventricular” (*juxtaventricular*) in “lesão parietal” (*parietal injury*) and in “lesão justa-ventricular” (*juxtaventricular injury*)), a feature captured by the WEs. Although it has few tokens on the test texts, they appear a lot on training data that enables a better learning of this class, which is why this NE class has high results. Value is related to numbers of therapeutic doses or to the results of diagnostic texts. Negation and Evolution are NE classes with many repeated tokens (see tables 4.2 and 4.3) and they are highly related to Condition and Results, e.g. “sem” (*without*) in “sem outras alterações” (*without other alterations*) and in “sem sintomas neurológicos” (*without neurological symptoms*) and “remissão” (*remission*) in “remissão dos sintomas” (*remission of the symptoms*) a characteristic caught by the CRF layer. DateTime is related with time, usually written using the same words and not depending on the author of the text (e.g. training texts contain “aos 60 anos” (*at 60 years old*) and “durante 21 dias” (*during 21 days*) and test texts have “aos 14 anos” (*at 14 years old*) and “durante o período da manhã” (*during the morning*)).

We were expecting better results for Condition, Test (T) and Therapeutics (THER) because they are too specific. Conditions use discriminatory affixes, e.g "dis" (difficulty), "exo" (out) and "meta" (alteration), and usually appear in the same contexts, near Anatomical Site, Test, Evolution or Therapeutics (e.g., "estenoses bilaterais" (*bilateral stenosys*), "Remissão dos sintomas" (*Remission of symptoms*), "terapêutica modificadora da doença" (*disease-modifying treatment*)).

Therapeutics has discriminatory affixes as well, since it covers active pharmaceutical ingredients ("azatioprina" or "dexametasona") or therapies to cure certain diseases ("imunoterapia" (*imunotherapy*), "corticoterapia" (*corticotherapy*) or "craniotomia descompressiva" (*decompressive craniectomy*). For the Test class, the F1-Score is highly influenced by the Prediction performance which means that there are several words from other NE classes, which were incorrectly classified as Test. We found that these words are mainly abbreviations. A possible explanation of such behaviour is due to the several abbreviations that the Test class contains in the training/validation dataset, e.g. "EEG", "RM-CE" and "SPECT".

As explained in section 4.1.1, the Results (R) could sometimes belong to the Condition class. Therefore, it was expected that the model would make some incorrect classifications in this class. However, we found that the major wrong classifications made by the classifier were on the texts written on items, e.g. "discreta anemia" in "...Intercorrências: Registo de crises epilepticas. Realização de SPECT ictal. Avaliação cognitiva. Análises: Hemograma com discreta anemia..." (*...Incidents: record of epileptic seizures. Realization of ictal SPECT. Cognitive Evaluation. Tests: Hemogram with mild anemia...*). Since the models were trained on full written clinical texts, the models are not well tuned for other types of text.

Although Route of Administration (RA) is a class with a large number of repeated tokens on the training/validation dataset, it presents low results on this test set because it has only three tokens on the entire set, i.e., each occurrence means a high percentage on its performance.

It is important to recall that the Genetics NE is not in the test set, and that the same set has only one token for Negation and Route of Administration, which explains the same relaxed and strict results for these NE classes.

Considering all the models it is interesting that for Negation and Route of Administration classes the results are very similar. These could be explained not only by the size of the occurrences, generally with only one token, but also with the specificity of the words which belong to them. There are also three classes where the dictionary-based model performed better than the CRF. These were not expected since the machine learning learn not only the words but also the transition probabilities. It certainly happens as a consequence of the transition rules which may not work well on texts written in items.

Average results for this independent dataset (Table 5.13) are about 10% lower than for the validation dataset. A possible reason for this is that the test set contains some admission notes and patient discharge letters, structured on items (e.g., origin, admission motive) and their description, which is different from the clinical cases in the validation dataset, described in a full paragraph that covers all related information. Furthermore, since they were not published, these texts were written less carefully, and therefore have some orthographic errors.

Table 5.13: Average results of all Models on independent Test Set

		Recall		**Precision**		**F1-Score**	
Model	Average	**Relaxed**	**Strict**	**Relaxed**	**Strict**	**Relaxed**	**Strict**
Residual Learning	**Micro**	72.55	61.43	73.94	62.20	71.21	61.82
BiLSTM-CRF		70.97	62.36	69.85	63.05	70.41	62.71
CRF		63.43	49.46	63.79	55.11	63.61	52.13
Dictionary-Based		44.91	41.37	75.68	48.81	56.37	44.78
Residual Learning	**Macro**	68.09	60.58	74.58	64.06	68.18	60.10
BiLSTM-CRF		67.97	61.74	67.13	61.17	65.45	60.10
CRF		59.15	49.11	69.59	62.15	56.81	50.12
Dictionary-Based		46.59	43.40	78.37	55.07	55.58	46.74
Residual Learning	**Weighted**	71.21	61.43	72.94	61.30	70.38	60.52
BiLSTM-CRF		70.97	62.36	69.75	61.91	68.52	61.10
CRF		63.43	49.46	70.07	60.77	61.39	51.31
Dictionary-Based		44.91	41.37	78.92	50.61	55.33	44.64

Once again, the deep learning models got better results than the CRF using both relaxed and strict evaluation. This is explained by the capacity of the combination of LSTM units, capable of getting valuable information from the texts, with the CRF layer capable of making relations between the different tags. However, the CRF model provides us information of how each token was labelled since, unlike deep learning models, it is not an black-box algorithm.

Considering the results of the dictionary-based model, we can conclude that the information written in the test dataset is not much different from the information written in the training dataset. This further supports that models of information extraction learned from open academic journals can indeed be used for extracting information from hospital documents.

6

Conclusion

This study assessed the performance of different machine learning models, namely Conditional Random Fields (CRF), Bidirectional Long Short-Term Memory (BiLSTM)-CRF and a model based on BiLSTM-CRF trained with residual learning method, in Named Entity Recognition (NER) from Portuguese clinical text. We can say that the main goals of this work were achieved, namely:

- We gathered and annotated a new dataset for Portuguese clinical text;
- We studied the best features for training NER models in clinical field;
- We learnt a Word Embedding (WE) model of Portuguese clinical text, also made publicly available and compared the performance of the previous approach when using this model with using general language WEs;
- We tested whether models trained with public clinical texts could be applied on clinical texts from hospitals, subjected to usage restrictions, due to privacy laws.

The first goal contributed with a new resource for Portuguese Natural Language Processing (NLP), as the datasets and WE models are publicly available in our GitHub repository[1]. This way, besides being used by us, the dataset may now be used by those willing to tackle the same problem, possibly with different approaches.

The in-domain WE model was trained with much lesser texts, but it lead to a higher performance. Although in a different language, this is in line with Griffis *et al.* [76], and confirms that, in the clinical domain, it should be better to train WE models exclusively with clinical texts, even if they are much less. This model is also publicly available in our GitHub repository, and may be used in a diversity of NLP tasks on clinical text.

[1] `https://github.com/fabioacl/PortugueseClinicalNER`

Furthermore, we also concluded that the deep learning model algorithms got the highest results, which is agreement with the state of the art results.

Although Tran et al. got better results when training their NER model using residual learning [73], our results show that it gets nearly the same performance as a BiLSTM-CRF model with just a single layer. However, our results have some limitations since we made the grid search globally, instead of for each layer independently, because of the computational cost that searching the hyperparameters for each layer would have.

Finally, we report a micro average F1-score of nearly 83% for the relaxed evaluation and about 75% for the strict evaluation on the validation dataset, and approximately 71% for relaxed evaluation and 63% for strict evaluation on an independent test dataset. The performance of the model in the independent test confirms that it is possible to train models for extracting information from hospital clinical texts without having direct access to them. In other words, Information Extraction (IE) models trained with public clinical cases extracted from journals are able to extract information from texts never seen before by the model. This is important, given the difficulty to access clinical texts from hospitals directly.

Furthermore, results of this study are useful for Hospital Neurology services, which may use this NLP tool for retrieving structured information from their raw reports. This will ease the population of databases, which will hopefully provide a more efficient way of analysing all the data, e.g., for finding relations between patient diseases and the therapeutics. An example of a clinical information extraction interface is presented in Appendice A - Figure A.1. It does not only labels the text with its Named Entity (NE) classes but also extracts the relevant information from each of them.

As future work, since machine learning models improve their classification with more data, it is important to increase the dataset size, which should then be used for learning better NER and WE models. Another approach that could improve the results would be using transfer learning for making WE models by training out-of-domain WE models with in-domain texts. Since there were some NE classes that got worse results, it is fundamental to check if they are really important or if they could be erased from the dataset and thus improve the results on the others.

It is also important to make a better grid search on both deep learning models to verify if it is possible to get better performances. Finally, it would be interesting to tackle relation extraction between NEs [96, 97], which, together with NER, would make it easier to summarize clinical reports.

Bibliography

[1] S. Folland, A. C. Goodman, and M. Stano, "Introduction," in *The Economics of Health and Health Care*, ch. 1, pp. 29–54, Pearson Prentice Hall Upper Saddle River, NJ, 8th ed., 2017.

[2] J. Oderkirk, "Readiness of Electronic Health Record Systems to Contribute to National Health Information and Research," *OECD Health Working Papers*, no. 99, pp. 1–80, 2017.

[3] M. Lamy, R. Pereira, J. C. Ferreira, J. B. de Vasconcelos, F. Melo, and I. Velez, "Extracting Clinical Information from Electronic Medical Records," in *International Symposium on Ambient Intelligence* (P. Novais, J. J. Jung, G. Villarrubia-González, A. Fernández-Caballero, E. Navarro, P. González, D. Carneiro, A. Pinto, A. T. Campbell, and D. Durães, eds.), Advances in Intelligent Systems and Computing, pp. 113–120, Springer, 2018.

[4] E. H. Shortliffe and M. J. Sepúlveda, "Clinical Decision Support in the Era of Artificial Intelligence," *Jama*, vol. 320, no. 21, pp. 2199–2200, 2018.

[5] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Shen, and Y. Wang, "Artificial Intelligence in Healthcare: Past, Present and Future," *Stroke and Vascular Neurology*, vol. 2, no. 4, pp. 230–243, 2017.

[6] D. A. Hashimoto, G. Rosman, D. Rus, and O. R. Meireles, "Artificial Intelligence in Surgery: Promises and Perils," *Annals of surgery*, vol. 268, no. 1, pp. 70–76, 2018.

[7] T. Cai, A. A. Giannopoulos, S. Yu, T. Kelil, B. Ripley, K. K. Kumamaru, F. J. Rybicki, and D. Mitsouras, "Natural Language Processing Technologies in Radiology Research and Clinical Applications," *Radiographics*, vol. 36, no. 1, pp. 176–191, 2016.

[8] K. Berezina, A. Bilgihan, C. Cobanoglu, and F. Okumus, “Understanding Satisfied and Dissatisfied Hotel Customers: Text Mining of Online Hotel Reviews,” *Journal of Hospitality Marketing & Management*, vol. 25, no. 1, pp. 1–24, 2016.

[9] F. J. Indurkhya, Nitin and Damerau, “Word Sense Disambiguation,” in *Handbook of Natural Language Processing*, ch. 14, pp. 315–338, Chapman & Hall/CRC, 2nd ed., 2010.

[10] L. Ferreira, A. J. S. Teixeira, and J. P. Cunha, “Information Extraction from Portuguese Hospital Discharge Letters,” *VI Jornadas en Technologia del Habla and II Iberian SL Tech Workshop*, no. January, pp. 39–42, 2010.

[11] A. Névéol, H. Dalianis, S. Velupillai, G. Savova, and P. Zweigenbaum, “Clinical Natural Language Processing in Languages other than English: Opportunities and Challenges,” *Journal of Biomedical Semantics*, vol. 9, p. 12, dec 2018.

[12] D. Jurafsky and J. H. Martin, “Introduction,” in *Speech and Language Processing*, ch. 1, pp. 1–16, Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2nd ed., 2009.

[13] P. Jackson and I. Moulinier, “Natural Language Processing,” in *Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization*, vol. 5 of *Natural Language Processing*, ch. 1, pp. 1–17, Amsterdam: John Benjamins Publishing Company, 2nd ed., jun 2007.

[14] A. Singhal, “Modern Information Retrieval: A Brief Overview,” *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, vol. 24, no. 4, pp. 35–43, 2001.

[15] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, “Open Information Extraction from the Web,” in *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI’07, (San Francisco, CA, USA), pp. 2670–2676, Morgan Kaufmann Publishers Inc., 2007.

[16] T. Luong, H. Pham, and C. D. Manning, “Effective Approaches to Attention-based Neural Machine Translation,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (Lisbon, Portugal), pp. 1412–1421, Association for Computational Linguistics, 2015.

[17] W. Xu, C. Callison-Burch, and C. Napoles, “Problems in Current Text Simplification Research: New Data Can Help,” *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 283–297, 2015.

[18] T. Wilson, J. Wiebe, and P. Hoffmann, “Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis,” in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, (Vancouver, British Columbia, Canada), pp. 347–354, Association for Computational Linguistics, 2005.

[19] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D., J. B., and K. Kochut, “Text Summarization Techniques: A Brief Survey,” *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 10, pp. 397–405, 2017.

[20] I. Santos, C. Laorden, B. Sanz, and P. G. Bringas, “Enhanced Topic-based Vector Space Model for Semantics-aware Spam Filtering,” *Expert Systems with Applications*, vol. 39, pp. 437–444, jan 2012.

[21] T.-H. Wen, M. Gasic, N. Mrkšić, P.-H. Su, D. Vandyke, and S. Young, “Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (Stroudsburg, PA, USA), pp. 1711–1721, Association for Computational Linguistics, sep 2015.

[22] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4960–4964, IEEE, mar 2016.

[23] R. Dale, “The Return of the Chatbots,” *Natural Language Engineering*, vol. 22, pp. 811–817, sep 2016.

[24] F. J. Indurkhya, Nitin and Damerau, “Information Extraction,” in *Handbook of Natural Language Processing*, ch. 21, pp. 511–532, Chapman & Hall/CRC, 2nd ed., 2010.

[25] D. Jurafsky and J. H. Martin, “Word & Transducers,” in *Speech and Language Processing*, ch. 3, pp. 45–83, Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2nd ed., 2009.

[26] D. Jurafsky and J. H. Martin, “Part-Of-Speech Tagging,” in *Speech and Language Processing*, ch. 5, pp. 123–173, Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2nd ed., 2009.

[27] D. Jurafsky and J. H. Martin, “Information Extraction,” in *Speech and Language Processing*, ch. 22, pp. 741–782, Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2nd ed., 2009.

[28] D. Jurafsky and J. H. Martin, “Parsing with Context-Free Grammars,” in *Speech and Language Processing*, ch. 13, pp. 431–464, Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2nd ed., 2009.

[29] M. Skeppstedt, M. Kvist, and H. Dalianis, “Rule-based Entity Recognition and Coverage of SNOMED CT in Swedish Clinical Text,” in *LREC*, pp. 1250–1257, 2012.

[30] A. Mykowiecka, M. Marciniak, and A. Kupść, “Rule-based Information Extraction from Patients’ Clinical Data,” *Journal of Biomedical Informatics*, vol. 42, no. 5, pp. 923–936, 2009.

[31] S. Gold, N. Elhadad, X. Zhu, J. J. Cimino, and G. Hripcsak, “Extracting Structured Medication Event Information from Discharge Summaries,” in *AMIA Annual Symposium Proceedings*, pp. 237–241, 2008.

[32] A. Henriksson, H. Dalianis, and S. Kowalski, “Generating Features for Named Entity Recognition by Learning Prototypes in Semantic Space: The Case of De-identifying Health Records,” in *2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 450–457, IEEE, nov 2014.

[33] Y. Wu, J. Xu, M. Jiang, Y. Zhang, and H. Xu, “A Study of Neural Word Embeddings for Named Entity Recognition in Clinical Text,” in *AMIA ... Annual Symposium proceedings. AMIA Symposium*, vol. 2015, pp. 1326–1333, 2015.

[34] B. Tang, H. Cao, Y. Wu, M. Jiang, and H. Xu, “Recognizing clinical entities in hospital discharge summaries using structural support vector machines with word representation features,” in *BMC medical informatics and decision making*, vol. 13, p. S1, BioMed Central, 2013.

[35] Y. Wu, M. Jiang, J. Xu, D. Zhi, and H. Xu, “Clinical Named Entity Recognition Using Deep Learning Models,” in *AMIA Annual Symposium proceedings. AMIA Symposium*, pp. 1812–1819, 2018.

[36] T. Chokwijitkul, A. Nguyen, H. Hassanzadeh, S. Perez, and L. Hospital, “Identifying Risk Factors For Heart Disease in Electronic Medical Records : A Deep Learning Approach,” in *Proceedings of the BioNLP 2018 workshop*, pp. 18–27, 2018.

[37] R. Srihari, “A hybrid approach for named entity and sub-type tagging,” in *Sixth Applied Natural Language Processing Conference*, (Seattle, Washington, USA), pp. 247–254, Association for Computational Linguistics, Apr. 2000.

[38] T. Rocktäschel, M. Weidlich, and U. Leser, "ChemSpot: a Hybrid System for Chemical Named Entity Recognition," *Bioinformatics*, vol. 28, pp. 1633–1640, 04 2012.

[39] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE 77*, vol. 77, no. 2, pp. 257–286, 1989.

[40] E. T. Jaynes, "Information Theory and Statistical Mechanics," *Physical review*, vol. 106, no. 4, pp. 620˜–630, 1957.

[41] R. Klinger and K. Tomanek, "Classical Probabilistic Models and Conditional Random Fields," Tech. Rep. TR07-2-013, Department of Computer Science, Dortmund University of Technology, 2007.

[42] A. McCallum, D. Freitag, and F. C. N. Pereira, "Maximum Entropy Markov Models for Information Extraction and Segmentation," in *Proceedings of the Seventeenth International Conference on Machine Learning*, (San Francisco, CA, USA), pp. 591–598, Morgan Kaufmann Publishers Inc., 2000.

[43] J. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," *Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 282–289, 2001.

[44] S. J. Russell and P. Norvig, "Probabilistic Reasoning over Time," in *Artificial Intelligence: A Modern Approach* (P. E. Limited, ed.), ch. 15, pp. 566–636, Pearson, 3rd ed., 2010.

[45] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.

[46] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv preprint arXiv:1301.3781*, pp. 1–12, 2013.

[47] J. Pennington, R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[48] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching Word Vectors with Subword Information,” *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[49] T. Mikolov, Q. V. Le, and I. Sutskever, “Exploiting Similarities among Languages for Machine Translation,” *arXiv preprint arXiv:1309.4168*, 2013.

[50] A. H. Marblestone, G. Wayne, and K. P. Kording, “Toward an Integration of Deep Learning and Neuroscience,” *Frontiers in Computational Neuroscience*, vol. 10, pp. 1–41, sep 2016.

[51] Y. Bengio, D.-H. Lee, J. Bornschein, T. Mesnard, and Z. Lin, “Towards Biologically Plausible Deep Learning,” *arXiv preprint arXiv:1502.04156*, pp. 1–10, 2015.

[52] I. Goodfellow, Y. Bengio, and A. Courville, “Introduction,” in *Deep Learning*, ch. 1, pp. 1–26, MIT Press, 2016.

[53] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.

[54] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, “Learning Hierarchical Features for Scene Labeling,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013.

[55] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Černocky, “Strategies for Training Large Scale Neural Network Language Models,” in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, pp. 196–201, IEEE, 2011.

[56] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, “Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups,” *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, nov 2012.

[57] M. K. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey, “Deep Learning of the Tissue-Regulated Splicing Code,” *Bioinformatics*, vol. 30, no. 12, pp. i121—-i129, 2014.

[58] H. Y. Xiong, B. Alipanahi, L. J. Lee, H. Bretschneider, D. Merico, R. K. C. Yuen, Y. Hua, S. Gueroussov, H. S. Najafabadi, T. R. Hughes, Q. Morris, Y. Barash, A. R. Krainer, N. Jojic, S. W. Scherer, B. J. Blencowe, and B. J. Frey, "The Human Splicing Code Reveals New Insights into the Genetic Determinants of Disease," *Science*, vol. 347, pp. 1–7, jan 2015.

[59] I. Goodfellow, Y. Bengio, and A. Courville, "Sequence Modeling: Recurrent and Recursive Nets," in *Deep Learning*, ch. 10, pp. 363–408, MIT Press, 2016.

[60] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[61] Y. Bengio, P. Simard, P. Frasconi, *et al.*, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[62] I. Goodfellow, Y. Bengio, and A. Courville, "Optimization for Training Deep Models," in *Deep Learning*, ch. 8, pp. 267–320, MIT Press, 2016.

[63] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[64] Ö. Uzuner, S. L. DuVall, B. R. South, and S. Shen, "2010 i2b2/VA Challenge on Concepts, Assertions, and Relations in Clinical Text," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 552–556, 2011.

[65] A. Stubbs and Ö. Uzuner, "Annotating Risk Factors for Heart Disease in Clinical Narratives for Diabetic Patients," *Journal of biomedical informatics*, vol. 58, pp. S78—-S91, 2015.

[66] M. Skeppstedt, M. Kvist, G. H. Nilsson, and H. Dalianis, "Automatic Recognition of Disorders, Findings, Pharmaceuticals and Body Structures from Clinical Text: An Annotation and Machine Learning Study," *Journal of Biomedical Informatics*, vol. 49, pp. 148–158, 2014.

[67] M. Rais, A. Lachkar, A. Lachkar, and S. E. A. Ouatik, "A Comparative Study of Biomedical Named Entity Recognition Methods based Machine Learning Approach," in *2014 Third IEEE International Colloquium in Information Science and Technology (CIST)*, pp. 329–334, IEEE, oct 2014.

[68] Y. Wang, Z. Yu, L. Chen, Y. Chen, Y. Liu, X. Hu, and Y. Jiang, "Supervised Methods for Symptom Name Recognition in Free-text Clinical Records

of Traditional Chinese Medicine: An Empirical Study," *Journal of Biomedical Informatics*, vol. 47, pp. 91–104, feb 2014.

[69] T. M. Luu, R. Phan, R. Davey, and G. Chetty, "Clinical Name Entity Recognition Based on Recurrent Neural Networks," *2018 18th International Conference on Computational Science and Applications (ICCSA)*, pp. 1–9, 2018.

[70] L. Kelly, L. Goeuriot, H. Suominen, A. Névéol, J. Palotti, and G. Zuccon, "Overview of the CLEF eHealth evaluation lab 2016," in *International Conference of the Cross-Language Evaluation Forum for European Languages*, pp. 255–266, Springer, 2016.

[71] K. Xu, Z. Zhou, T. Hao, and W. Liu, "A Bidirectional LSTM and Conditional Random Fields Approach to Medical Named Entity Recognition," in *International Conference on Advanced Intelligent Systems and Informatics*, pp. 355–365, 2018.

[72] I. Jauregi Unanue, E. Zare Borzeshi, and M. Piccardi, "Recurrent Neural Networks with Specialized Word Embeddings for Health-domain Named-entity Recognition," *Journal of Biomedical Informatics*, vol. 76, no. June, pp. 102–109, 2017.

[73] Q. Tran, A. MacKinlay, and A. Jimeno Yepes, "Named Entity Recognition with Stack Residual LSTM and Trainable Bias Decoding," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, (Taipei, Taiwan), pp. 566–575, Asian Federation of Natural Language Processing, 2017.

[74] A. Prakash, S. A. Hasan, K. Lee, V. Datla, A. Qadir, J. Liu, and O. Farri, "Neural Paraphrase Generation with Stacked Residual LSTM Networks," in *Proceedings of {COLING} 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, (Osaka, Japan), pp. 2923–2934, The COLING 2016 Organizing Committee, 2016.

[75] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, IEEE, jun 2016.

[76] D. Newman-Griffis and A. Zirikly, "Embedding Transfer for Low-Resource Medical Named Entity Recognition: A Case Study on Patient Mobility," in *Proceedings of the BioNLP 2018 workshop*, pp. 1–11, 2018.

[77] P. V. Q. de Castro, N. F. F. da Silva, and A. da Silva Soares, "Portuguese Named Entity Recognition Using LSTM-CRF," in *Computational Processing of the Portuguese Language - 13th International Conference, PROPOR 2018, Canela, Brazil, September 24-26, 2018, Proceedings*, pp. 83–92, 2018.

[78] N. Hartmann, E. Fonseca, C. Shulby, M. Treviso, J. Rodrigues, and S. Aluisio, "Portuguese Word Embeddings: Evaluating on Word Analogies and Natural Language Tasks," *Proceedings of the 11th Brazilian Symposium in Information and Human Language Technology*, pp. 122–131, 2017.

[79] C. D. Santos and B. Zadrozny, "Learning Character-level Representations for Part-of-speech Tagging," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1818–1826, 2014.

[80] C. dos Santos and V. Guimarães, "Boosting Named Entity Recognition with Neural Character Embeddings," *Proceedings of the Fifth Named Entity Workshop*, pp. 25–33, 2015.

[81] H. Dalianis, M. Hassel, and S. Velupillai, "The Stockholm EPR Corpus-Characteristics and Some Initial Findings," in *Proceedings of the 14th International Symposium for Health Information Management Research*, (Kalmar), pp. 243–249, 2009.

[82] S. P. de Neurologia, "Sinapse," in *Publicações da Sociedade Portuguesa de Neurologia*, vol. 17 of *1*, (Lisbon), pp. 1–196, Sociedade Portuguesa de Neurologia, 2017.

[83] S. P. de Neurologia, "Sinapse," in *Publicações da Sociedade Portuguesa de Neurologia*, vol. 17 of *2*, (Lisbon), pp. 1–184, Sociedade Portuguesa de Neurologia, 2017.

[84] J. Klatt, H. Feldwisch-Drentrup, M. Ihle, V. Navarro, M. Neufang, C. Teixeira, C. Adam, M. Valderrama, C. Alvarado-Rojas, A. Witon, and Others, "The EPILEPSIAE database: An Extensive Electroencephalography Database of Epilepsy Patients," *Epilepsia*, vol. 53, no. 9, pp. 1669–1676, 2012.

[85] R. Rodrigues, H. G. Oliveira, and P. Gomes, "NLPPort: A Pipeline for Portuguese NLP (Short Paper)," in *7th Symposium on Languages, Applications and Technologies (SLATE 2018)* (P. R. Henriques, J. P. Leal, A. M. Leitão, and X. G. Guinovart, eds.), vol. 62 of *OpenAccess Series in Informatics (OASIcs)*, (Dagstuhl, Germany), pp. 18:1—-18:9, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018.

[86] E. F. Tjong Kim Sang and F. De Meulder, “Introduction to the CoNLL-2003 Shared Task: Language-independent Named Entity Recognition,” in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL ’03, (Stroudsburg, PA, USA), pp. 142–147, Association for Computational Linguistics, 2003.

[87] L. d. S. Ferreira, *Medical Information Extraction in European Portuguese.* PhD thesis, Universidade de Aveiro, 2011.

[88] T. Mikolov, E. Grave, P. Bojanowski, P. Gupta, and A. Joulin, “Learning Word Vectors for 157 Languages,” in *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, pp. 3483–3487, 2018.

[89] R. Rehurek and P. Sojka, “Software Framework for Topic Modelling with Large Corpora,” in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, (Valletta, Malta), pp. 45–50, ELRA, 2010.

[90] G. Bouma, “Normalized (Pointwise) Mutual Information in Collocation Extraction,” *Proceedings of the Biennial GSCL Conference 2009*, pp. 31–40, 2009.

[91] F. Sha and F. Pereira, “Shallow Parsing with Conditional Random Fields,” in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pp. 213–220, Association for Computational Linguistics, 2003.

[92] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *arXiv preprint arXiv:1412.6980*, pp. 1–13, 2014.

[93] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding Deep Learning Requires Rethinking Generalization,” *arXiv preprint arXiv:1611.03530*, 2016.

[94] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[95] Y. Benjamini and Y. Hochberg, “Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 57, pp. 289–300, jan 1995.

[96] S. Sahu, A. Anand, K. Oruganty, and M. Gattu, “Relation Extraction from Clinical Texts using Domain Invariant Convolutional Neural Network,” in *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*,

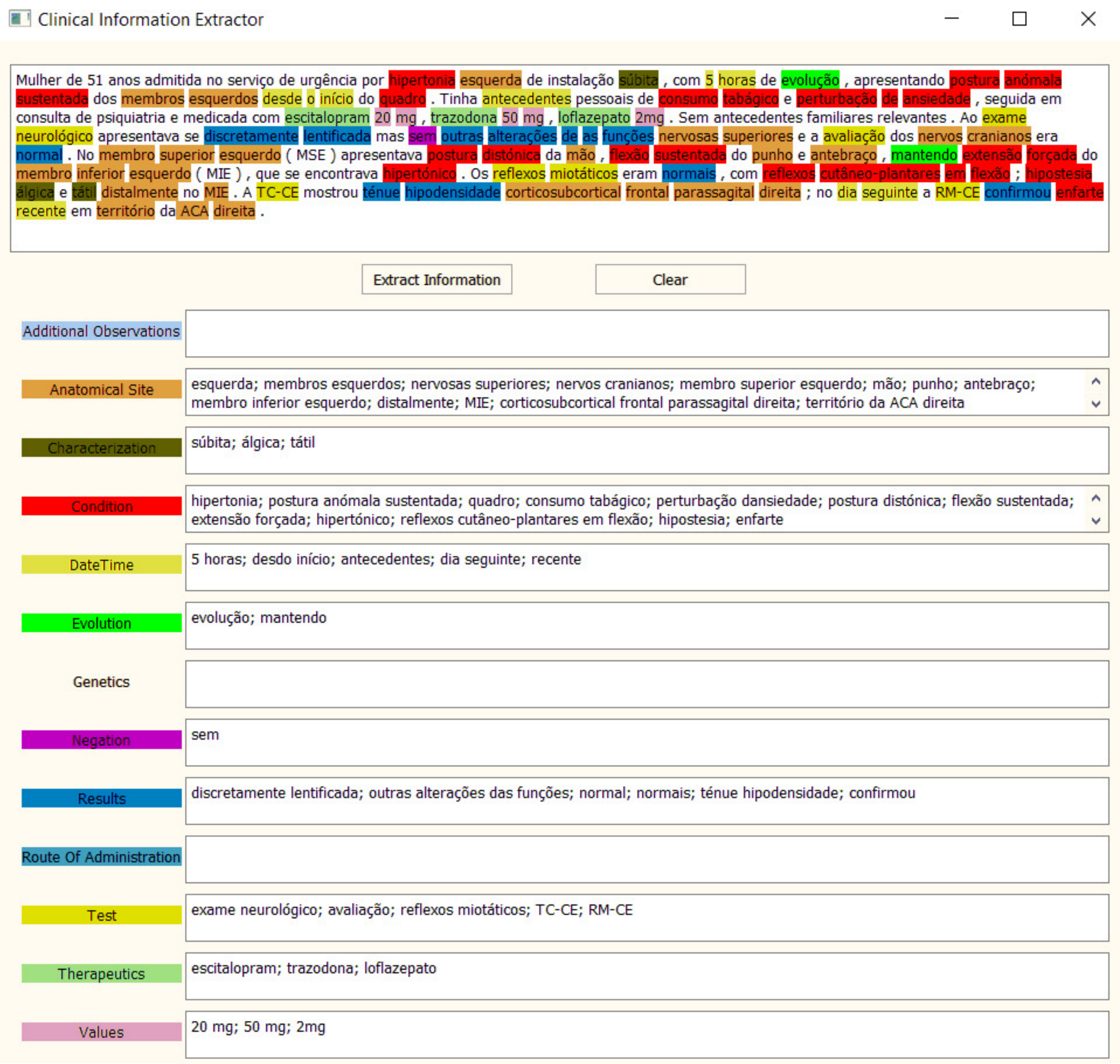
(Berlin, Germany), pp. 206–215, Association for Computational Linguistics, 2016.

[97] S. Collovini, G. Machado, and R. Vieira, "A Sequence Model Approach to Relation Extraction in Portuguese," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, (Portorož, Slovenia), pp. 1908–1912, European Language Resources Association (ELRA), may 2016.

Appendices

A

Clinical Information Extraction Interface



Mulher de 51 anos admitida no serviço de urgência por hipertonia esquerda de instalação súbita , com 5 horas de evolução , apresentando postura anómala sustentada dos membros esquerdos desde o início do quadro . Tinha antecedentes pessoais de consumo tabágico e perturbação de ansiedade , seguida em consulta de psiquiatria e medicada com escitalopram 20 mg , trazodona 50 mg , loflazepato 2mg . Sem antecedentes familiares relevantes . Ao exame neurológico apresentava se discretamente lentificada mas sem outras alterações de as funções nervosas superiores e a avaliação dos nervos cranianos era normal . No membro superior esquerdo (MSE) apresentava postura distónica da mão , flexão sustentada do punho e antebraço , mantendo extensão forçada do membro inferior esquerdo (MIE) , que se encontrava hipertónico . Os reflexos miotáticos eram normais , com reflexos cutâneo-plantares em flexão ; hipostesia álgica e tátil distalmente no MIE . A TC-CE mostrou ténue hipodensidade corticosubcortical frontal parassagital direita ; no dia seguinte a RM-CE confirmou enfarte recente em território da ACA direita .

Additional Observations	
Anatomical Site	esquerda; membros esquerdos; nervosas superiores; nervos cranianos; membro superior esquerdo; mão; punho; antebraço; membro inferior esquerdo; distalmente; MIE; corticosubcortical frontal parassagital direita; território da ACA direita
Characterization	súbita; álgica; tátil
Condition	hipertonia; postura anómala sustentada; quadro; consumo tabágico; perturbação dansiedade; postura distónica; flexão sustentada; extensão forçada; hipertónico; reflexos cutâneo-plantares em flexão; hipostesia; enfarte
DateTime	5 horas; desdo início; antecedentes; dia seguinte; recente
Evolution	evolução; mantendo
Genetics	
Negation	sem
Results	discretamente lentificada; outras alterações das funções; normal; normais; ténue hipodensidade; confirmou
Route Of Administration	
Test	exame neurológico; avaliação; reflexos miotáticos; TC-CE; RM-CE
Therapeutics	escitalopram; trazodona; loflazepato
Values	20 mg; 50 mg; 2mg

Figure A.1: Clinical Information Extraction Interface.