Adverse drug reactions detection in clinical notes

Edson Alejandro Florez Suarez

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Détection des Effets Indésirables des Médicaments dans les Notes Cliniques

Adverse Drug Reactions Detection in Clinical Notes

Edson Alejandro FLÓREZ SUÁREZ

Laboratoire I3S

Présentée en vue de l'obtention du grade de docteur en Informatique **d'**Université Côte d'Azur
Dirigée par : Michel Riveill
Soutenue le : 01 juillet 2020

Devant le jury, composé de :
Andrea G.B. Tettamanzi, Professeur, Université Côte d'Azur, *Examinateur*
Pascal Staccini, Professeur, Université Côte d'Azur, *Examinateur*
Christian Retore, Professeur, Université de Montpellier, *Rapporteur*
Marc Cuggia, Professeur, Université Rennes 1, *Rapporteur*

FRANCELABS SEARCH EXPERTS

i3S sophia antipolis

Région Provence-Alpes-Côte d'Azur

Abstract

The Information Extraction from clinical notes provides relevant information to identify adverse side effects in post-marketing surveillance of medications (Pharmacovigilance), which is more difficult to discover by traditional medical studies since patients are taking several treatments at the same time. In recent years, data mining techniques have allowed to discover knowledge stored in big datasets, such as the clinical records collected by hospitals throughout patient's life. The goal of this work is identify adverse side effects caused by treatments. Then, we have to identify relations between medications and Adverse Drug Events (ADE) entities, which is called Adverse Drug Reaction relation. This problem is divided Named Entity Recognition (NER) and Relation Extraction tasks. Nowadays, supervised approaches based on Deep Learning and Machine Learning algorithms solve this problem in the state of the art. These supervised systems require rich features in order to learn efficient models during training, therefore, we focus on building comprehensive word representations (the input of the neural network), using character-based word representations and word representations. The proposed representation improves the performance of the baseline model, and the final model reached the performances of state of the art methods. Then we have extracted contextual information through Deep Learning models and other different features obtained from the relations, in order to identify the Adverse Drug Reaction relations. The proposed model improved the overall accuracy and the extraction of Adverse Drug Reaction compared to the baseline, indicating the effectiveness of combining Deep Learning models and extensive feature engineering.

Keywords: Deep Learning, Information Extraction, Adverse Drug Reaction, Adverse Drug Event, Clinical Notes

Résumé

L'extraction d'information de textes médicaux fournit des renseignements très utiles pour identifier les effets indésirables dans la surveillance après consommation (Pharmacovigilance), qui sont plus difficiles à découvrir à travers des études médicales typiques puisque les patients prennent plusieurs traitements en même temps. Récemment, les techniques de Data Mining ont permis de découvrir les connaissances enregistrées dans de grands ensembles de données, comme les dossiers cliniques collectés par les hôpitaux tout au long de la vie du patient. L'objectif de cette thèse est d'identifier les effets indésirables causés par les traitements. Pour cela, nous devons extraire les relations entre les médicaments et Adverses Drug Events (ADE), qui est la relation de réaction indésirable des médicaments. Ce problème est divisé en tâches de reconnaissance d'entités nommées (NER) et d'extraction de relations. Aujourd'hui, les approches supervisées basées sur des algorithmes de Deep Learning et Machine Learning résolvent ce problème dans l'état de l'art. Les méthodes supervisées ont besoin de caractéristiques riches afin d'apprendre des modèles efficaces au cours de la formation, par conséquent, nous nous concentrons sur la construction de représentations de mots larges (l'entrée du réseau neuronal), nous utilisons des représentations de mots basées sur des caractères et des représentations de niveau de mots. La représentation proposée améliore la performance du modèle de référence et le modèle final a atteint les performances des méthodes de pointe. Ensuite, nous avons extrait des informations contextuelles à travers des modèles de Deep Learning, afin d'identifier les réactions indésirables aux médicaments. Le modèle proposé a amélioré la précision globale et l'extraction des réactions indésirables aux médicaments obtenu avec le modèle de base, ce qui indique l'efficacité de combiner des modèles de Deep Learning et une vaste ingénierie des caractéristiques.

Mots-clés: Deep Learning, Extraction d'Information, Adverse Drug Reaction, Adverse Drug Event, texte médicaux

Acknowledges

Dr. Michel Riveill proposed the research topic at the I3S laboratory, with the support of MD. Pascal Staccini from the CHU de Nice (UCA). I am very grateful to Professor Michel Riveill for his full dedication and leadership to this PhD thesis, with the continuous advising of Professor Frederic Precioso (I3S Laboratory, UCA).

This work was partly funded by the French government labelled PIA program under its IDEX UCA JEDI project (ANR-15-IDEX-0001). The Provence-Alpes-Cote d'Azur (PACA) region and the France Labs company finance the doctoral work in Laboratoire d'Informatique, Signaux et Systèmes de Sophia-Antipolis (I3S) - UMR7271 - UCA CNRS.

Many thanks to Cédric Ulmer, CEO of France Labs, the industrial partner of this PhD work, and its researcher PhD. Romaric Pigetti for his collaboration in the publications of this thesis.

Thanks to MD. Virginie Lacroix-Hugues and MD. David Darmon from the Faculté de Médecine of UCA, for their specialized advising and provision of the PRIMEGE database (Regional Information Platform in General Medecine).

Thanks to researcher Clement Jonquet from Montpellier from LIRMM (Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier) at the Université de Montpellier, for the collaboration to use the BioPortal (LIRMM) annotator of French biomedical ontologies and terminologies.

Thanks to researcher Chloé Cabot from CISMef (Catalog and Index of French Language Medical Sites) at the CHU Hôpitaux de Rouen, for the support to use the ECMT annotator (Extracting Concepts with Multiple Terminologies).

We also work with Master Students in Computer Science at UCA. Eliane Birba Delwende for pre-processing of PRIMEGE clinical notes using Deep Learning (Delwende, 2018), project started with Athip Ponna and Upasana Biswas. Jacqueline Neef for data analysis of PRIMEGE database.

I also would like to thank to Dr. Carlos Jaime Barrios, professor at Universidad Industrial de Santander (UIS), Colombia, for his support during the whole process of this PhD thesis.

Thanks to my colleagues and friends in the Côte d'Azur, for sharing many unforgettable moments. To Nice la Belle (*Nissa La Bella*), my home for three years.

Finally, to my family who encouraged me continuously from an ocean of distance.

Contents

Introduction **1**
1.1 Motivation ... 1
1.2 Problem ... 2
1.3 Objective ... 4
1.4 Contributions ... 4
1.5 Outlines ... 5
Information Extraction for Adverse Drug Reaction Detection **8**
2.1 Introduction ... 8
2.2 Formal definition of the problem ... 8
2.3 Word Representations ... 10
2.4 Methods of Solution ... 14
2.5 Supervised approach for Adverse Drug Reaction Detection ... 16
2.6 Conclusion ... 18
Named Entity Recognition in Clinical Notes **20**
3.1 Introduction ... 20
3.2 Related Work ... 20
3.3 Supervised Approach ... 23
3.3.1 Datasets ... 24
3.3.2 Supervised Learning Models ... 26
3.4 Results and Discussion ... 35
3.4.1 MADE Results ... 35
3.4.2 Results with Dataset in Spanish ... 40
3.4.3 Important issues in clinical notes ... 42
3.5 Conclusion ... 43
Relation Extraction in Clinical Notes **45**
4.1 Introduction ... 45
4.2 Related Work ... 45
4.3 Relation Extraction Model ... 48
4.3.1 Candidate Generation ... 49
4.3.2 Feature Extraction ... 49
4.3.3 Training ... 50

4.3.4 Transfer Learning 51
4.4 Experiments 52
4.4.1 Dataset 52
4.4.2 Experimental settings 53
4.5 Results and Discussion 54
4.6 Conclusion 57
Real Life Scenario **60**
5.1 Introduction 60
5.2 Raw Clinical Data 60
5.3 Clinical Notes Pre-processing 63
5.4 Automatic annotators based on ontologies 64
5.5 NER for Medical Text in French 68
5.6 Conclusion 74
Conclusions and Future Work **76**
6.1 Conclusions 76
6.2 Future Work 78
Appendices **81**
Bibliography **83**

List of Tables

3.1: Methods for ADE extraction 23

3.2: Performance of models with MADE dataset 36

3.3: Performances of models for NER task in MADE Challenge (test set) 37

3.4: Performance by category on test dataset of our best model in MADE challenge ... 37

3.5: Confusion matrix between entities 38

3.6: Performance of models 39

3.7: Performance by category on test dataset of our best model 39

3.8: NER task results in MADE Challenge (Strict Evaluation) 40

3.9: Overall performance for NER task on test set 41

3.10: F1 score by category on test dataset 41

3.11: Performance by category on test dataset of the best model 42

4.1. Relation Extraction results in MADE Challenge, NER Task 47

4.2: Performance of LSTM+Features model with Test dataset of MADE Challenge ... 54

4.3: Performance metrics for the relation extraction task 55

4.4: Performance (F1) with Test dataset of MADE Challenge 56

4.5: Comparison with MADE challenge task 3 57

5.1: UMLS type of entities identified by ECMT in PRIMEGE notes 67

5.2: Number of annotations of original and corrected PRIMEGE clinical notes 67

5.3: Number of annotations by category 69

5.4: Results for entity recognition task in CLEF eHealth 2015 (Névéol, et al., 2015) .. 70

5.5: Results for entity recognition task in CLEF eHealth 2016 (Névéol, et al., 2016) .. 70

5.6: Performance for plain entity recognition on MEDLINE test set 72

List of Figures

2.1: Example of entities and ADR relation in a sentence 10

2.2: One-hot encoding for N words 11

2.3 CBOW and Skip-gram models (Mikolov, Chen, Corrado, & Dean, 2013) 12

2.4: Pipeline for full ADR detection 16

2.5: Clinical Note sample with annotations in XML format 17

3.1: Sequence label scoring of sentence with linear-chain CRF 27

3.2: Comprehensive word representation 29

3.3: Baseline model based on LSTM for NER on MADE dataset 31

3.4: Final model for sequence tagging 32

3.5: Model based on LSTM and CRF for NER on MADE dataset 32

3.6: Model based on LSTM and Attentional model for NER on MADE dataset 34

3.7: Attentional model (Luong, Pham, & Manning, 2015) 35

4.1: Relation Extraction module 48

4.2: Relation Extraction model 51

4.3: BERT model for Sentence Pair classification tasks. 52

4.4: Number of annotations for every type of relation in MADE dataset 53

4.5: Pipeline for Joint task 57

5.1: Description of data collected in PRIMEGE 61

5.2: Entity–relationship model of PRIMEGE database 62

5.3: Number of elements in PRIMEGE 63

5.4: ECMT annotation through web service 65

5.5: Samples of annotations of PRIMEGE 68

5.6: The best model for NER in MEDLINE dataset 71

5.7: Full model for ADR detection on PRIMEGE dataset 73

6.1: Language models with size in millions of parameters 78

CHAPTER 1

Introduction

1.1 Motivation

The detection of adverse side effects of medications is a complex problem in post-marketing surveillance, which belongs to the field of Pharmacovigilance. Nowadays, patients are taking several treatments at the same time, therefore their bodies are under Drug Drug Interaction (DDI) that could yield undesirable effects. Drug Drug Interactions are changes in a drug's effect when the drug is taken together with one or more drugs. It can delay, decrease or increase the action of the drugs, or even cause adverse effects. Therefore, the motivation of this research is the emergence of new adverse drug effects, which are more difficult to detect by traditional medical studies and experiments since patients are taking more medications at the same time today.

In recent years, data mining techniques for Information Extraction have allowed to discover knowledge stored during many years in big datasets, such as the clinical records collected by hospitals throughout patient's life. The data mining is necessary to exploit the huge amounts of clinical data available, in order to discover new side effects that are affecting people's health. Electronic Health Records save the patient's health in structured records but also in rich unstructured text, such as clinical notes that are written by general practitioners and medical specialists, who use medical vocabulary and jargon like medication and disease names. Some medical centers collect anonymous clinical data and provide it for research purpose.

Clinical notes contains medical observations, symptoms, diagnoses, reasons of encounter, etc., that also provides important information for surveillance of adverse effects of medications (Hauben & Bate, 2009). The future work could provide tools to support the doctor's decisions during the medication prescription, taking into account the potential adverse side effects in real-time Pharmacovigilance (Drug Safety) (Wang, Hripcsak, Markatou, & Friedman, 2009). For this purpose, several automated methods

of Information Extraction have been proposed in the literature, which try to overcome the specific challenges related to information extraction applied to clinical data.

1.2 Problem

The problem consists in detecting side effects of treatments. If an adverse effect occurred in presence of any drug, it is called Adverse Drug Event (ADE), and then if the ADE is consequence of taking a drug, it is considered or classified as an Adverse Drug Reaction (ADR). The data mining of Electronic Health Records is necessary to discover that specific information. This is an Information Extraction task between named entities in Natural Language Processing field. Previous approaches to the problem merely identified ADE mentions, without look for relations with treatments.

Moreover, previous approaches are dictionary-based and rule-based, thus they cannot be generalized, and the systems are inflexible to recognize ambiguous events (such as ADE) in different context of sentences, because the dictionaries (or terminologies) do not have all possible forms of the entities. For instance, the Adverse Reaction Terminology (WHO-ART) collects a list of common vocabulary of Adverse Drug Events, but this terminology has no links with medications that could produce it. Although such approaches have been accurate for detecting explicit entities, such as Drug's names, which are proper names well defined in specific terminologies or ontologies. Instead, supervised learning approaches can be trained to predict entities from any domain defined by the annotated data.

This problem is complex due to different reasons, for example, the system receives unknown vocabulary such as new medications or chemicals products, or ambiguous entities that cannot been defined completely by dictionaries. Therefore, it is necessary an appropriate representation of clinical data, to represent words that do not have standard representations. The approaches could fail to distinguish between different events that include same words, for instance the word "fever" could be consider as ADE or symptom of diseases according to the context. They are just entities in this initial procedure, with no relation recognized with other type of entities (e.g. relations between ADE and medications).

The last systems related to Adverse Drug Reactions are based on Machine Learning and Deep Learning algorithms. Mostly, these systems address the sequential problem of

Named Entity Recognition, such as the work done by (Nikfarjam, Sarker, O'connor, Ginn, & Gonzalez, 2015), without obtain relations between entities. Machine Learning algorithm such as Support Vector Machines (SVMs) was used for the identification of adverse effects mentions of drugs with a dataset of ADE annotations in medical text (Gurulingappa, Mateen-Rajpu, & Toldo, 2012). ADR mentions were detect using Deep Learning models (Huynh, He, Willis, & Rüger, 2016) such as CNN (Convolutional Neural Network), and RNN (Recurrent Neural Network) is more specialized on this type of sequential problems (Liu, et al., 2017). RNN is limited due to the vanishing gradients problem (Bengio, Simard, & Frasconi, 1994), then another RNN architecture known as Long Short-Term Memory (LSTM), reduced this problem using a short memory connection along the input sequence. LSTM has been used to exploit the long-term dependencies inside word sequences to increase the accuracy of this Named Entity Recognition tasks (Jagannatha & Yu, 2016).

It is important to feed the neuronal network with an appropriate input representation (Chiu & Nichols, 2016), in order to improve accuracy of LSTM, for example, a vector representation like the Skip-gram word embedding in (Jagannatha & Yu, 2016). We could also improve the accuracy with additional features, such as character-level features and concatenate character and word representations inspired by the work of Chiu et. al. (Chiu & Nichols, 2016).

Results of works for ADE detection were collected in the review made by (Sarker, et al., 2015), where Machine learning and Deep Learning algorithms are outstanding, although the comparison is not precise because each author used different datasets. Therefore, recently, some challenges have been organized in this research field, to allow comparison of systems executed under the same conditions.

Previous works for detection of relations between medical entities (like medications and ADE) (Jagannatha, Liu, Liu, & Yu, 2019) were approaches grouped into rule-based, lexicon-based, and supervised learning mostly (Jagannatha, Liu, Liu, & Yu, 2019). Nowadays, the works use supervised learning due to the high accuracy of Machine Learning and Deep Learning methods. The systems are based on Deep Learning models such as Bidirectional Long Short-Term Memory with Attention layer (Dandala, Joopudi, & Devarakonda, 2018), and Machine Learning algorithms such as Random Forests (Chapman A. B., Peterson, Alba, DuVall, & Patterson, 2018; Magge, Scotch, &

Gonzalez-Hernandez, 2018) and Support Vector Machines (Xu, Yadav, & Bethard, 2018).

The researchers in (Munkhdalai, Liu, & Yu, 2018) take the previous words with a fixed window size of both candidate entities as an input of the LSTM layer. In the LSTM model proposed by (Dandala, Joopudi, & Devarakonda, 2019), the input of the LSTM layer is the sentences between the entities of the relation, included the sentences in which the entities appeared. Some works were evaluated in the MADE Challenge (Yu, Jagannatha, Liu, & Liu, 2018), where a system based on Random Forest archives the best result (Chapman A. B., Peterson, Alba, DuVall, & Patterson, 2018). See more details of the state-of-the-art in Chapters 3 and 4.

1.3 Objective

The goal of the thesis is to develop models for automatic detection of Adverse Drug Reactions in clinical data. We address the problem by a supervised approach divided into Named Entity Recognition and Relation Extraction tasks, with a model able to learn patterns during training with annotated data from clinical notes. We base the model on recently proposed Deep Learning methods, and we try to exploit contextual information and different features of clinical notes for classification of entities into categories defined by labelled data, to finally extract relations between the entities with the trained model. Therefore, given a clinical note as input to the trained model, the model returns pairs of entities and their relations such as the Adverse Drug Reaction relation between Adverse Drug Events and medications.

1.4 Contributions

The global contribution of this dissertation is the full data pipeline for identification of entities and its relations, using supervised models focused on Adverse Drug Reactions (ADR). The identification of relations between ADE and medications is the most challenging task for detecting Adverse Drug Reactions in clinical notes. Most of the existing works only have performed the Named Entity task, so their models do not get any relation. Instead, we developed the full procedure to extract relations between entities, using the annotated relations provided in datasets for supervised learning.

First, we explore the impact of character embedding to classify named entities involved in the ADR relation (ADE and Drug entities). We have implemented models based on Long Short-Term Memory (LSTM) with a wide word representation (with character embedding, word embedding and Part of Speech), which improves the performance of LSTM by itself. We validated our approach through the participation of international NLP (Natural Language Processing) challenges, for evaluation and comparison of official results with state–of–the–art methods executed under the same conditions. This work was published in Proceedings of NLP Challenges for Detecting Drug and Adverse Drug Events from Electronic Health Records (MADE 2018) (Yu, Jagannatha, Liu, & Liu, 2018), with the collaboration of PhD. Romaric Pigetti (Florez, Precioso, Riveill, & Pighetti, 2018), researcher of France Labs company, the industrial partner of this doctoral work. Additionally, we test the generalization skill of the model for identification of medical-related entities using other language (Spanish dataset), with the gold standard dataset provided by PharmacoNER Challenge, and the results were published in the Workshop on BioNLP Open Shared Tasks by (Agirre, et al., 2019).

Second, we extract information from external features to enrich a model based on Deep Learning, for identification of relations between the entities. The combination of that features vector and contextual knowledge is effective to detect relations, because the external features provide other important type of information to improve the accuracy of the baseline model. We also used the dataset provided by the recent challenge as benchmarking to compare with state-of-the-art models (Yu, Jagannatha, Liu, & Liu, 2018). This work was published in Proceedings of the International Conference on Natural Language Processing (Florez, Precioso, Pighetti, & Riveill, 2019). Finally, we can perform the full task (NER and Relation Extraction) with a pipeline to detect Adverse Drug Reactions, given only raw clinical notes as input of the pre-trained model.

1.5 Outlines

The rest of the dissertation is organized as follows:

Chapter 2 describes the main concepts related to this thesis. It presents state-of-the-art approaches for the problem specifically focused on the medical field. We define the formal problem and introduce the main methods to solve it. Finally, we review the word

representations available for the sequential input, and we present the overall view of the model to address the full problem, with modules for Named Entity Recognition and Relation Extraction tasks.

Chapter 3 reviews the state-of-the-art for supervised approaches for Named Entity Recognition. We describe the proposed supervised models to identify entities in clinical notes, based on Machine Learning and Deep Learning algorithms. The experiments were carried out with gold-standard datasets of challenges to evaluate the models in same conditions as other research teams (during the challenges), and we present the experimental setup to allow the reproduction of experiments.

Chapter 4 describes the full Relation Extraction model, with the preliminary Candidate Generation and Feature Extraction of entity pairs, before the Deep Learning based method for identification of relations between entities. There we compare against the most recent state-of-the-art methods to validate the results, methods mostly based on Machine Learning algorithms.

Chapter 5 describes a real life scenario for Adverse Drug Reaction detection in raw clinical notes in French. There we explain the issues of raw data provided directly from the source with minor pre-processing, where we shows the necessity of de-noising procedures. Then the annotations were made using Dictionary-based methods available for medical data in French. Finally, we could compare these type of approaches with our model for Named Entity Recognition in a gold-standard corpus in French.

Chapter 6 summarizes the conclusions of the thesis and we propose perspectives to continue the research in future work.

CHAPTER 2

Information Extraction for Adverse Drug Reaction Detection

2.1 Introduction

EHR (Electronic Health Records) storages the patient health in both structured records and unstructured text such as clinical notes, which are written by general practitioners and medical specialists with medical vocabulary, e.g. medication and chemical names. Some medical centers collect and publish anonymous data for research purpose.

Clinical notes contain medical observations, symptoms, diagnoses, reasons of encounter, etc. it also provides important information for surveillance of adverse side effects. A tool for adverse side effects detection can support the doctor's decisions during the medication prescription in real-time Pharmacovigilance (Drug Safety) (Hauben & Bate, 2009).

In the Supervised approach for automatic detection of entity mentions in clinical notes, we learn a model from annotated data, and then we try to identify and annotate the medical entities found in the raw clinical notes. Deep Learning methods can do the automatic extraction of adverse events from large number of Clinical Notes.

2.2 Formal definition of the problem

In the supervised data mining approach, an unlabeled sequence of words have to be classified into some category or None. Usually, it is a multi-class classification problem where the model is trained to classify in more than two classes, the labeled entities are composed by one or more words.

Formally, given the sets of words X and its labels Y for the words sequence $x = (x_1, x_2, \dots, x_t)$ of length t, get a classification function $f : X \rightarrow Y$ that assigns every word $x_i \in X$ to its corresponding label $y_i \in Y$. Then the corpus is divided into training and

testing set, the training set for leaning the classification model from text and its labels. The test set is unlabeled texts that are used to evaluate the accuracy of the model in predicting the target labels.

The categories are selected according to the interests/necessities of the domain annotators (applications). The scientific community publishes corpus for medical research purpose, for example, the dataset made with Medical Case Reports for detecting sentences only with ADE or non ADE (Gurulingappa, et al., 2012), and datasets with finer-grain annotations like the QUAERO French Medical Corpus, which has ten categories (Névéol, Grouin, Leixa, Rosset, & Zweigenbaum, 2014) such as Anatomy, Devices, Chemical and Drugs, Disorders, etc.

The Named Entity Recognition (NER) tasks belong to Natural Language Processing domain (Information Extraction precisely), where we need to identify objects (one or many words) that belong to some predefined categories. The common categories are proper names for places, persons and organizations (Poibeau & Kosseim, 2001), numbers such as quantities and percentage, and temporal expressions such as dates. Categories with complex vocabulary of specific domains such as chemical products, diseases and genes.

We try to establish if a side effect has been caused by any treatments, therefore, it is necessary to identify the medicine and side effect categories. For example, we only identify the event *internal bleeding* in the sentence *"The patient has internal bleeding …"*. Then if we find any medication in the same context, we can consider this event as an Adverse Drug Event (ADE). ADE is an adverse event that happens simultaneously when the patient takes a medication, whether it is identified as a cause of the event or not. For instance in the sentence *"The patient has <ADE>internal bleeding</ADE> secondary to <Drug>warfarin</Drug>."*, where an ADE and Drug entities are labelled. There given the pair of entities (X_1, X_2) and the set of labels *Y* for relations (None included), we get a classification function $f : (X_1, X_2) \rightarrow Y$ that assigns every possible pair (X_1, X_2) to its corresponding relation $y_i \in Y$.

The Relation Extraction tasks begins after the identification of medications and ADE using NER models, then we have to consider the context of the full sentences to know if there is a relation between the ADE and Drug entities (see Fig. 2.1). If the Adverse Drug Event was caused by the drug, it is a relation called ADR (Adverse Drug

Reaction), as in the explicit statement "*the patient has internal bleeding secondary to warfarin*".

Entities: *"The patient has <ADE>internal bleeding</ADE> ... <Drug>warfarin</Drug>."*
Relation: *"The patient has <ADE>internal bleeding</ADE> secondary to <Drug>warfarin</Drug>."*

Figure 2.1: Example of entities and ADR relation in a sentence

The evaluation metrics reported by most of the authors are F1 score, Precision and recall. Precision and recall take into account the True Positives (TP) or number of correct predictions of the gold standard evaluation data. The unlabeled elements or negative samples belong to the None class (True Negatives).

Precision is focused on False Positives (FP) predictions. Precision is the ratio of True Positives predictions to the total positive predictions made by the model:

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

Recall is focused on False Negatives (FN) or number of samples of the gold standard evaluation data that the model did not predict. Recall is the ratio of True Positives predictions to all the samples of the gold standard evaluation data:

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

F1 Score is the weighted average of Precision and Recall, then F1 score takes into account both False Positives and False Negatives. Therefore, it is considered the main metric to evaluate the models:

$$F1\ Score = \frac{2(Recall * Precision)}{Recall + Precision}$$

The classification problems based on word sequences typically need informative representations as input, instead the original word without any type of features.

2.3 Word Representations

The input of word sequences should provide relevant information for the classification, thus there is necessary good representations or features. The sequence of words (or tokens) are replace by any representation or vectors obtained through

algorithms such as Bag of Words (BoW), CBOW (Continuous BoW), N-grams, Skip-Gram, Word vectors and FastText. The first following models are count-based machine learning applied to NLP tasks, which storage the representations as a vocabulary in lookup tables.

We can cluster words in classes, where similar words share the same class (or parameters) for the purpose of generalization, because **Word Classes** method assumes that similar words appear in similar contexts. Therefore, each word of the vocabulary is mapped to a single class, for example one class for cities (Berlin, Paris, Rome) and other class for countries (Germany, France, Italy).

One-hot (1-of-N) representations is a simple way to encode categorical data, such as words, using only discrete values 1 and 0. N is the size of the vocabulary in the 1-of-N encoding, then we will have a matrix of N x N, and every word *w* receives only its corresponding 1 (see Fig 2.2). The main problem One-hot representations is the high dimension of the matrix, then it is used in our work only to encode the set of classes (labels) that contains less than 50 elements.

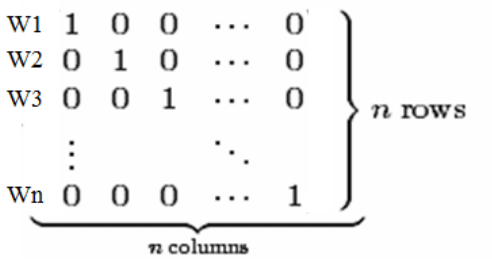


Figure 2.2: One-hot encoding for N words

Bag-of-words (BoW) for documents or sentences is the number of occurrence of each word in the given sample, i.e. it is the sum of one-hot codes without take into account the order of words. Then the input sequence is represented as a vector of words, which can be considered as the N-gram model with N=1. For example in "*The patient has internal bleeding secondary to warfarin, another patient has neuropathy due to the same medication", w*e would get the BoW:

{"The":2, "patient":2, "has":2, "bleeding":1, "secondary":2, "to":2, "warfarin":1, "another":1, "neuropathy":1, "due":1, "same":2, "medication":1}

The **N-gram** models are able to store contextual information, using the same frequency method of BoW, however it splits the sequences of text in more than one word (unigram), to conform bigrams (term of two words) or N-grams of N words. For example, "internal bleeding" would be a bigram and the model will count the number of occurrences in the text.

Continuous Bag-of-Words models (CBoW) add inputs from words within short window (the context) to predict the current word (Mikolov, Chen, Corrado, & Dean, 2013). The weights for different positions are shared in the weight matrix of the hidden layer (projection), then it is computationally more efficient but it cannot model n-grams. **Skip-gram** is a CBoW variation (the inverse) that try to find word representations for predicting the surrounding words (the context) of the target word in a sentence during training (see Fig. 2.3). Larger training context results in more training examples and thus can lead to a higher accuracy (Mikolov, Chen, Corrado, & Dean, 2013). For example with a window size 2 in the sentence: *The patient has internal <Target>bleeding</ Target> secondary to warfarin***,** CBoW would take the context words (vectors) of the target word as input for training, i.e. "has internal" and "secondary to" are used to predict "bleeding" (the label). On the contrary, "bleeding" would be the input in Skip-Gram, and the model predict the context words "has internal" and "secondary to". Then the model back propagates to minimize the prediction error, from the output layer to the weight matrix using Cross Entropy as loss function.

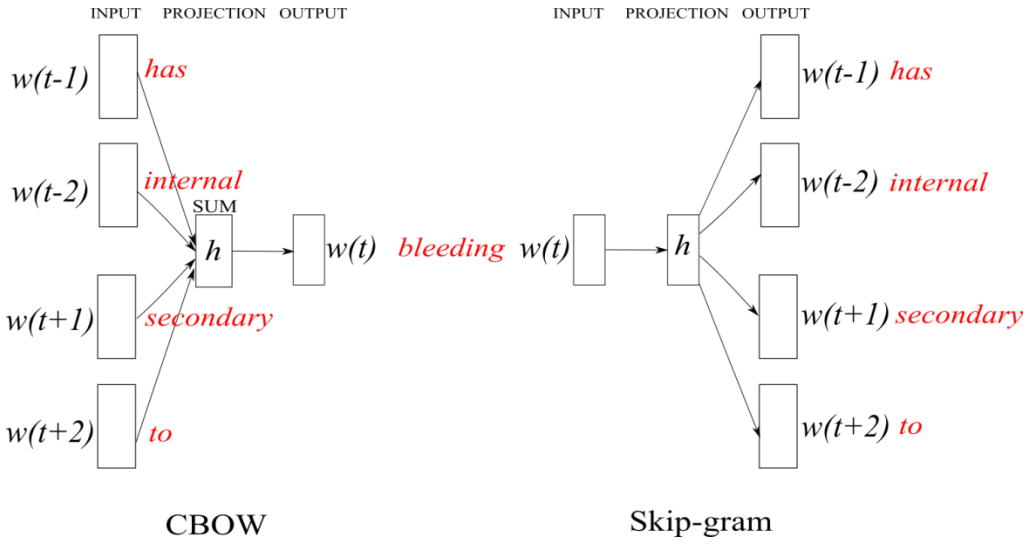


Figure 2.3 CBOW and Skip-gram models (Mikolov, Chen, Corrado, & Dean, 2013)

The **Word Vectors** (or embeddings) have some similar properties to word classes, but word vectors capture many degrees of similarity (Paris is similar to Rome, but also to France) and also capture linguistic properties such as gender (queen and king, aunt and uncle) (Mikolov, Yih, & Zweig, 2013). The word embeddings (word vectors) are dense vectors in the matrix (usually the matrix of weights) between the input and hidden layer. The model will learn a continuous representation of words represented by a real valued vector compressed in a low N-dimensional space (Bengio, Ducharme, Vincent, & Jauvin, 2003), where words that appear in similar contexts are mapped to nearby vectors by the parameterized function. Word embeddings are learned on large unlabeled datasets through different algorithms (Collobert & Weston, 2008). The classifier algorithm have many outputs as there are words in the vocabulary, where the previous word (encoded as one-hot) is used to predict the current word by going through hidden layer.

The **word2vec** project implements CBOW and Skip-Gram for training embedding (Mikolov, Chen, Corrado, & Dean, 2013), with the extensions of the original Skip-gram model using sub-sampling of frequent words improves accuracy of less frequent words representations, and a variant of Noise Contrastive Estimation for training the Skip-gram model that results in better vector representations for frequent words compared to Hierarchical Softmax.

FastText extends the continuous skip-gram model of word2vec (Mikolov, Chen, Corrado, & Dean, 2013) by adding subword information, in order to obtain representations of rare words by a sum of its character n-grams (Bojanowski, Grave, Joulin, & Mikolov, 2017). This method is efficient to text representation learning on large corpora. FastText outperforms CBOW and Skip-gram models of word2vec in almost all datasets of the state-of-the-art models for word representations (Mikolov, Grave, Bojanowski, Puhrsch, & Joulin, 2017).

Word vectors can be also trained in a layer of full neural network language model, as projection vector of the input, but it involves dense matrix multiplications and supervised training that is not efficient (Mikolov, Chen, Corrado, & Dean, 2013). Pre-trained word representations are provided as vectors into a lookup table (dictionary shape). The pre-trained word vectors provide generalization for systems trained with limited amount labelled data in tasks such as Named Entity Recognition (Sienčnik,

2015). Then we represent words through Word2Vec and FastText embeddings to add worthy generalization features to the classifiers.

2.4 Methods of Solution

The main approaches for Information Extraction are rule-based models and supervised learning. **Rule-based** approaches are usually based on handcrafted rules for sentences. These systems are difficult to build because it requires extensive domain knowledge. The rules are provided as language patterns using grammatical and syntactic (e.g. Part of Speech POS and word precedence), thus rule-based are inflexible to understand all the different contexts in which entities appear in the sentences. Rule-based systems are implemented in combination with **dictionary-based** approaches to increase the accuracy (Budi & Bressan, 2003). They are used commonly in domains with high formalism that facilitates the creation of terminologies, e.g. biomedical annotators based in ontologies (like BioPortal, ECMT and LIRMM). However, dictionary-based approaches are limited to one domain because they are only capable to detect entities that are in the dictionary.

Supervised Learning methods consist in training algorithms that pick up statistical patterns in labelled data, in order to learn discriminative features and apply them to unseen data. Supervised models learn to classify specific categories defined in annotated samples, in consequence, these models require large annotated datasets and they have to be adapted to every domain. Most of supervised models are based on Machine Learning algorithms such as Hidden Markov Models (HMM) (Rabiner, 1989), Decision Trees (Rokach & Maimon, 2008), Support Vector Machines (SVM) (Joachims, 1998), Conditional Random Fields (CRF) (Lafferty, McCallum, & Pereira, 2001), they have been used in works for clinical entities recognition such as SVM (Tang, Cao, Wu, Jiang, & Xu, 2013) and CRF (Settles, 2004). Recently, Deep Learning algorithms such as Recurrent Neural Networks (RNN) have been state of the art in biomedical NLP task (Li, Jin, Jiang, Song, & Huang, 2015). The most recent approaches will be review in the following chapters (see Chapters 3 and 4).

Semi-Supervised Learning (SSL) models are able to use un-labelled data in addition to labeled data for learning (Nadeau D. , 2007), such as semi-supervised model based on CRF that trains on both type of data simultaneously (Liao &

Veeramachaneni, 2009). The main problem to work in new domains is the need of specific annotations, then semi-supervised learning reduce the annotation efforts of training data, although supervised models still get more accuracy. In contrast to labeled data, unlabeled data is available in huge amounts from sources like Wikipedia.

Unsupervised methods have been used for entity recognition some years ago, however they still need the support of domain dictionaries or handcrafted rules. An unsupervised system for Named-Entity Recognition creates large lists of entities for a given type of entity or semantic class such as car brands or cities, and then it uses heuristics to perform named-entity classification (Nadeau, Turney, & Matwin, 2006). Other unsupervised method to biomedical named-entity recognition does not need rules or training data, the system uses term collection extracted from terminologies for each target entity (disorders, treatments, etc.), boundary detection to keep entities correlated with noun phrases, and a classifier to predict the semantic category of candidate entities in clinical notes and biomedical data (Zhang & Elhadad, 2013).

Dictionary-based approaches are not suitable for Relation Extraction tasks. Relation Extraction also includes distant supervision based techniques and some few techniques which jointly extract entities and relations (Pawar, Palshikar, & Bhattacharyya, 2017). Distant supervision (Mintz, Bills, Snow, & Jurafsky, 2009) does not require labelled data, instead it needs a large semantic database for automatically obtaining relation labels. This method is based on heuristics as any sentence might express a relation if contains both entities of the relation, so the database contains entity pairs for each relation type. They train a multi-class logistic classifier using lexical, syntactic and entity type features.

Jointly extraction models for entities and relations, such as graphical models approach, train local independent (entity and relation) classifiers with dependencies between entities and relations, which are encoded by a Bayesian Belief network (Roth & Yih, 2004). It is a directed acyclic graph where entities and relations are represented as nodes in two different layers, each relation node has two incoming edges from its entity nodes. They provide a feature vector for the sentences, with constraints encoded through the conditional probabilities estimated from the entities and relations labelled corpus or set manually. The best reported F-measure for joint modelling is still low (on dataset ACE2004) (Pawar, Palshikar, & Bhattacharyya, 2017).

2.5 Supervised approach for Adverse Drug Reaction Detection

We propose a full method for identification of entities and their relations through supervised approach in clinical notes (see Fig. 2.4). The approach is based on Machine Learning and Deep Learning algorithms. We could collect data from clinical corpus made publicly available from the research community. Clinical notes divide into sequences of words are the input for the pipeline (final trained model).

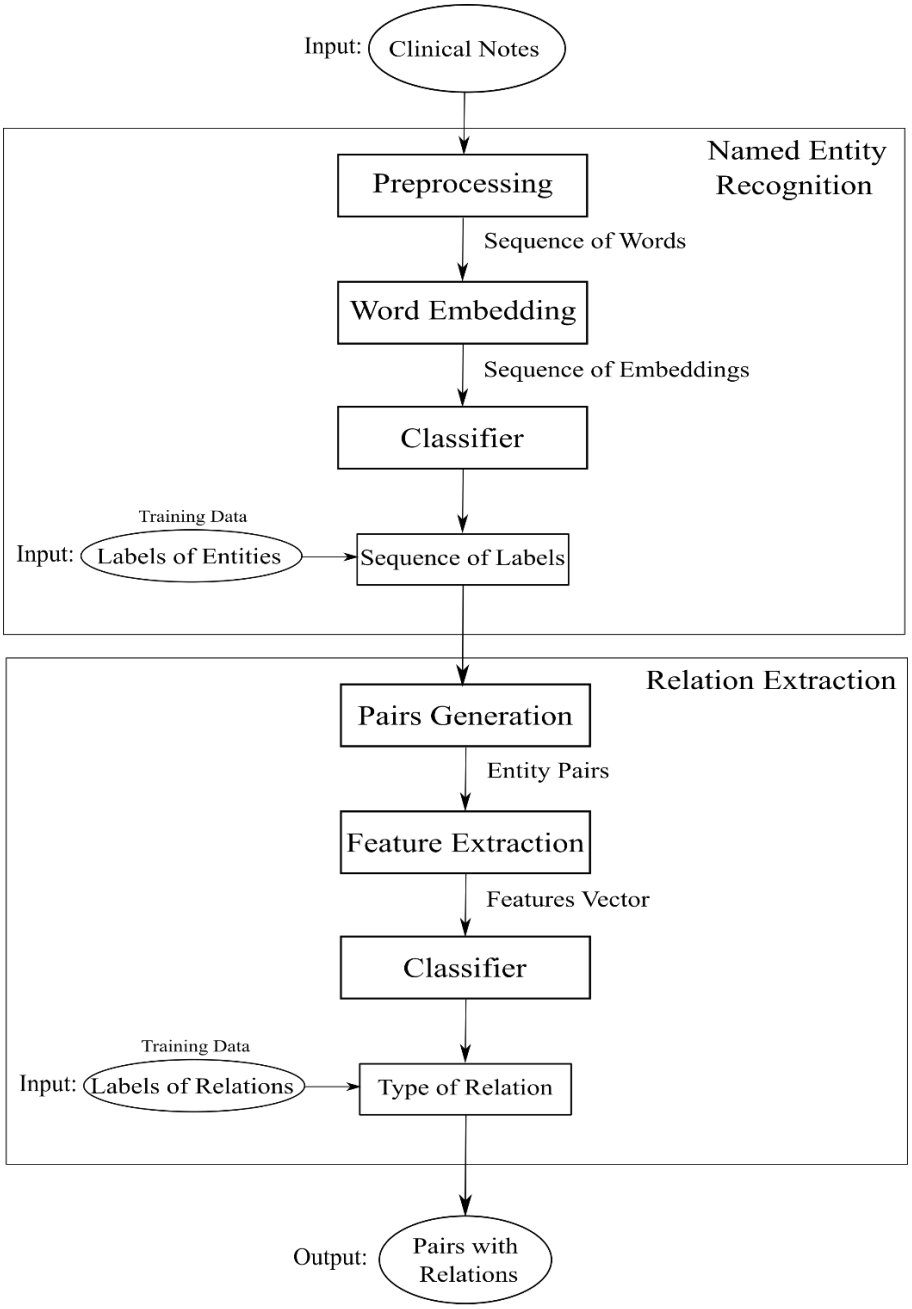


Figure 2.4: Pipeline for full ADR detection

First, we preprocess and tokenize the raw clinical notes (see Fig. 2.5), each document is split into sequence of words (with N tokens length) until the end of sentences in the document. Sentences longer than N tokens were cropped to size, and shorter sentences were pre-padded with masks to indicate where the last word is. The sequences of words need pre-processing with a regular expression tokenizer into individual word and special character tokens in lower case to match with the word embedding.

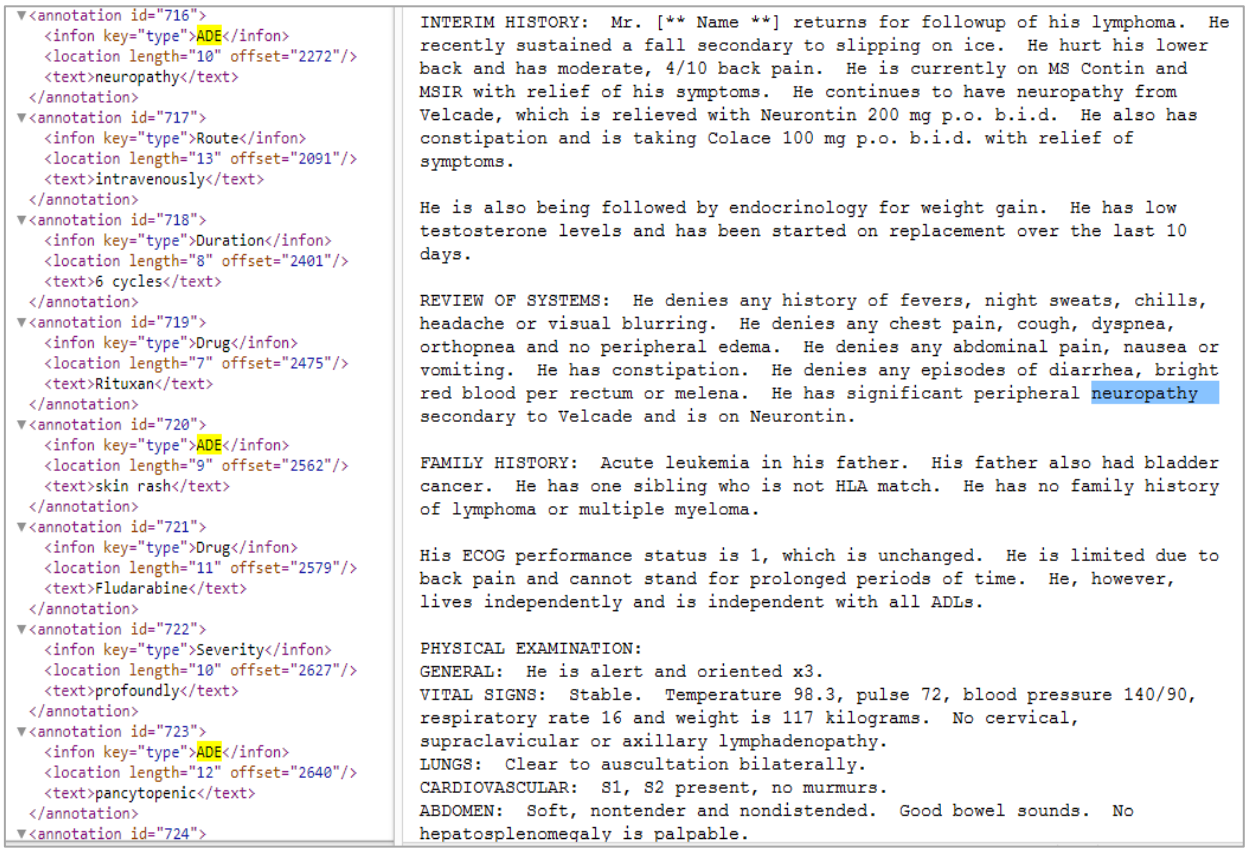


Figure 2.5: Clinical Note sample with annotations in XML format

We add features and word representation in the input layer. Then Named Entity Recognition module is performed and the output is the predicted entities, it is another input to the Relation Extraction module in the pipeline. The supervised learning is based on annotations for both entities and relations presented in clinical notes, for example, the annotations of entities:

[Begin, End, Text, Label, #Entity]

[2272, 2282,"neuropathy", ADE, 716]

[2295, 2302,"Velcade", Drug, 717]

Annotations of relations:
[#Entity1, #Entity2, Type, #Relation]
[716, 717, Adverse, 1]

The final output is the relation between the entities. The description of each module of this model is in Chapters 3 and 4.

2.6 Conclusion

Clinical notes contain rich information such as medical observations, diagnoses, medications, etc, and the information required for our surveillance of adverse side effects. The information extraction on clinical notes can be performed by Supervised Learning methods that overcome the limitations of other methods such as dictionary based models. In the supervised approach for detection of entity mentions, we learn a model from annotated data, then it try to identify and annotate the medical entities in raw clinical notes.

We propose a full method for identification of entities and their relations through supervised approach in clinical notes. The input of word sequences can provide relevant information for the supervised model, the state of the art of word embeddings shows FastText and word2vec (Skip-gram) as good representations for words, so we used them in our models.

CHAPTER 3

Named Entity Recognition in Clinical Notes

3.1 Introduction

Extracting medical events from clinical notes provides relevant information for surveillance of adverse side effects, because clinical notes contain richer information about patient health than structured records. Patients are often subject to multiple treatments, which may be the cause of adverse effects. Therefore, it is necessary to establish if an Adverse Drug Event (ADE) has occurred after taking medicines. ADE refers to any adverse event occurring at the time a drug is taken, whether it is identified as a cause of the event or not. In case one can establish a relation between the ADE and the drug, then the relation is considered as an Adverse Drug Reaction (ADR), which is a Relation Extraction (RE) task. Deep Learning models could improve the identification of possible ADEs in real-time Pharmacovigilance (Drug Safety).

In the Supervised Learning approach for automatic detection of entity mentions in clinical notes, we learn a model from annotated data, then we try to identify and annotate the medical entities found in raw clinical notes. We can find ADE mentions in clinical notes provided in EHR (Electronic Health Records). These notes contain mentions of medical entities like medications, ADE (Adverse Drug Event), symptoms, etc. These terms have to be identified in the classification problem known as Named Entity Recognition (NER). A named entity is a term (one or many words) that can be annotated with a label (tagging) if it belongs to any predefined category. The next subsection presents related works to NER in medical domain.

3.2 Related Work

Adverse Drug Event detection has been performed with systems based on Machine Learning and Deep Learning algorithms recently. Machine Learning algorithm such as

Conditional Random Fields (CRFs) is used for ADR extraction (Liu, et al., 2017), CRF can take context (around the current word) into account for sequence modeling, it takes every neighbour word in a fixed window of words (Nikfarjam, Sarker, O'connor, Ginn, & Gonzalez, 2015). Support Vector Machines (SVMs) is other Machine Learning algorithm used commonly for NER. Gurulingappa et. al. (Gurulingappa, Mateen-Rajpu, & Toldo, 2012) built a system for the identification and extraction of potential adverse events of drugs with SVM. Their dataset is an ADE corpus from MEDLINE (Medical Literature Analysis and Retrieval System Online) case reports annotated manually. The corpus contains annotations for the mentions of drugs, ADE, and relations between drugs and medical conditions representing clear adverse reactions (relation drug-cause-condition).

The CLEF (Cross-Language Evaluation Forum) eHealth Evaluation Lab provides system performance for NER, the Task 1b in CLEF 2015 (Névéol, et al., 2015), using the QUAERO French Medical Corpus (Névéol, Grouin, Leixa, Rosset, & Zweigenbaum, 2014). It has ten categories for annotations of medical entities, with data collected from the EMEA (European Medicines Agency) documents and titles of research articles indexed in the MEDLINE database. A Dictionary-based concept recognition system overcame CRF and SVM classifiers in CLEF 2015 on the MEDLINE corpus (Névéol, et al., 2015), according to the Exact Match metric, which considers a term (word or group of words that have a label) as correctly classified only if all the words in the term received the correct label.

Deep learning models like CNN (Convolutional Neural Network) are used to detect the presence of ADR (Huynh, He, Willis, & Rüger, 2016), such as the binary classification problem on two medical datasets from Twitter and case reports (Gurulingappa, Mateen-Rajpu, & Toldo, 2012). Overall, CNN appears to perform better compared to other more complex CNN variants that have a RNN (Recurrent Neural Network) layer (Gurulingappa, Mateen-Rajpu, & Toldo, 2012). However, CCNA (Convolutional Neural Network with Attention) is better on the dataset of case reports. In overall, results of the case reports are better than results obtained with the Twitter dataset of medical domain. Tweets contain a lot of ill-grammatical sentences and short forms that hinders the performances (Huynh, He, Willis, & Rüger, 2016), which highlights the importance of de-noising the data.

The adverse event detection problem focused on clinical notes is a sequential problem, and RNN is specialized for it because at time step *t*, the recurrent node takes as input the outputs produced by the previous state. RNN models were limited to make separate classifications at every time step on an input sequence (Liwicki, Graves, Fernàndez, Bunke, & Schmidhuber, 2007), but they face the problem of vanishing gradients (Bengio, Simard, & Frasconi, 1994), instead another RNN architecture known as Long Short-Term Memory (LSTM), reduces the impact of this problem using a short memory connection along the sequence. LSTM was designed to take into account the long-time dependencies between relevant inputs of the sequence. LSTM has been applied to sequential problems such as Handwriting Recognition (Liwicki, Graves, Fernàndez, Bunke, & Schmidhuber, 2007) and Named Entity Recognition (Jagannatha & Yu, 2016). LSTM exploits the long term label dependencies for sequence labelling in clinical text, e.g. in "*the patient has internal bleeding* (ADE) *secondary to warfarin* (Medication)", the sentence contains an ADR relation between ADE and Medication entities, and the label for ADE is strongly related to the label prediction of Medication. Then "*internal bleeding*" is tagged as ADE using information of Medication label, which is stored in the memory of LSTM cells.

LSTM was used with an annotated corpus of English Electronic Health Records (EHR) from cancer patients in (Jagannatha & Yu, 2016), with labels for several medical entities (like Adverse Drug Event, drug name, dosage, etc.) and relations between entities. The best model in (Jagannatha & Yu, 2016) is the Approximate Skip Chain CRF-RNN network (see Table 3.1), which implements a CRF algorithm after the bidirectional LSTM output, and a Skip-gram word embedding calculated using unlabelled data from PubMed, English Wikipedia and unlabeled EHR corpus (called MADE dataset), these EHRs are not used in the annotated dataset for training and test. This network has a high accuracy for Drug name detection, but a low accuracy for ADE, probably because the dataset in unbalanced and has less ADE samples and the confusion between ADEs and categories with the same vocabulary (such as SSD).

Table 3.1 shows results of NER algorithms dedicated to ADE detection, some of them were collected in the review article made by (Sarker, et al., 2015), but each author used different datasets so it is not possible to make comparisons in same conditions. This review shows that Machine learning and Deep Learning algorithms are outstanding at this task, but those results were obtained with different datasets, making the

comparison somewhat unfair. However, the best result used the same dataset, (Huynh, He, Willis, & Rüger, 2016) and (Gurulingappa, Mateen-Rajpu, & Toldo, 2012) (last lines of Table 3.1), the SVM model in (Gurulingappa, Mateen-Rajpu, & Toldo, 2012) obtained slightly better results than CNNA on Recall, Precision and F-score.

Study	Method	Size	Recall	Prec.	F1
(Nikfarjam & Gonzalez, 2011)	Lexical pattern-matching	1200	0.66	0.70	0.68
(Nikfarjam, Sarker, O'connor, Ginn, & Gonzalez, 2015)	Supervised learning via Conditional Random Fields (CRFs)	1559	0.78	0.86	0.82
(Jagannatha & Yu, 2016)	Skip-CRF-Approx. (Bi-LSTM-CRF)	1154	0.83	0.81	0.82
(Huynh, He, Willis, & Rüger, 2016)*	CNNA (Convolutional Neural Network with Attention)	2972	0.84	0.82	0.83
(Gurulingappa, Mateen-Rajpu, & Toldo, 2012)*	SVM (Support Vector Machines)	2972	**0.86**	**0.89**	**0.87**

Table 3.1: Methods for ADE extraction
Note: *Systems using the same dataset

LSTM model has shown to be appropriate on the state of the art for sequential problems. However, in order to improve performance, it is important to feed the network with an appropriate input representation (an embedding) (Chiu & Nichols, 2016). This representation replaces each unique word with a dense vector representation, which tries to provide closer vectors among word synonyms or related words. In (Jagannatha & Yu, 2016) the embedding layer values used were initialized using a Skip-gram word embedding and unlabelled data from three open access corpus mentioned before. We could also improve the precision of LSTM with additional features for its input, such as character-level features from each word extracted using CNN or LSTM (Liu, et al., 2017), and then concatenate character and word representations inspired by the work of Chiu et. al. (Chiu & Nichols, 2016).

3.3 Supervised Approach

This section presents the experiments to validate our model for NER in clinical notes. We describe the datasets, models (Sections 3.3.1 and 3.3.2) and present results (Section 3.3.3).

3.3.1 Datasets

We participated in two challenges in order to validate and compare the results on gold-standard corpus. The first NLP Challenges for Detecting Drug and Adverse Drug Events from Electronic Health Records, the MADE challenge (Yu, Jagannatha, Liu, & Liu, 2018) with text in English, and the first NER task on chemical, drug, gene/protein mention recognition from clinical case studies in Spanish, which is called PharmaCoNER challenge (Agirre, et al., 2019). We also studied a medical NER task with text in French, the CLEF eHealth Evaluation Lab 2015 (Névéol, et al., 2015) that used the QUAERO French Medical Corpus (MEDLINE source) abovementioned (Névéol, Grouin, Leixa, Rosset, & Zweigenbaum, 2014), which requires specific embedding for text in French to classify ten types of medical entities (see Subsection 5.5).

- **MADE Dataset**

The dataset for ADE research was provided by the MADE Challenge (Yu, Jagannatha, Liu, & Liu, 2018). MADE challenge is focused on extracting fine grained structured information related to Drug Safety (Xu, Yadav, & Bethard, 2018). This dataset was created with 1092 EHR notes from 21 cancer patients (Jagannatha & Yu, 2016), which contains annotations for nine entity types: ADEs, indications, other signs and symptoms, medication, dosage, route, frequency, duration, severity. It also provides relations among those medical entities for the Relation Extraction task, e.g. the Adverse relation between Medication and ADE entities. The dataset contains 876 clinical notes for training and 213 clinical notes for test dataset established by the MADE challenge (see Table 3.1). The full 1089 clinical notes have 79003 annotations, about 86% of annotations for training, and average of 800 Words/Document approx.

Annotations	Training	Test
ADE	1509	431
SSLIF	**34056**	**5328**
drug	13507	2395
indication	3168	636
frequency	4148	658
duration	765	133
route	2278	389
dosage	4893	801
severity	3374	534
Total Ann.	67698	11305
Number of files	876	213

Table 3.1: Distribution of annotations by entity in MADE dataset

- **Dataset in Spanish**

We compare another dataset for NER in other language, a dataset in Spanish provided by PharmaCoNER (Pharmacological Substances, Compounds and proteins and Named Entity Recognition) organization (Agirre, et al., 2019). They hold the first NER task on chemical, drug and gene/protein from medical notes (clinical case studies) in Spanish, for identifying particular problems of non-English corpus and develop dedicated NER tool for other languages. The Spanish clinical notes are a manually classified collection of clinical case sections gathered from Spanish Clinical Case Corpus (SPACCC) (Agirre, et al., 2019), with annotations related to the medical domain. These clinical cases cover multiple medical topics, including oncology, urology, cardiology, diseases, etc., which is important to obtain a diverse collection of chemicals and medications. Clinical cases from other fields such as psychology or historical forensics were removed. The dataset contains annotations of 1000 clinical cases, which includes four entity types (*Normalizables, No_Normalizables, Proteinas* and *Unclear*):

– "*Normalizables*": 4426 mentions of chemicals that can be manually normalized to a unique concept identifier (mostly SNOMED-CT).

– "*No_Normalizables*": 55 mentions of chemicals that could not be normalized manually to a unique concept identifier.

– "*Proteinas*": 2291 mentions of proteins and genes that include also peptides, peptide hormones and antibodies.

– "*Unclear*": 159 cases of general substance class mentions of clinical and biomedical relevance, including general treatments, chemotherapy programs, a predefined set of

general substances (e.g. *Estragón, Melanina, Vaselina, Alcohol, Tabaco, Cannabis* and *Gluten*), etc.

These Named Entity Recognition (NER) annotations allow training for tagging medical entities found in clinical notes (raw text files). The corpus contains 16504 sentences (average of 16.5 sentences per clinical case) and 396988 words (average of 396.2 words per clinical case).

3.3.2 Supervised Learning Models

The main structure of the model has three layers, Embedding layer, Bi-LSTM (Bidirectional LSTM) layer and CRF layer, in the middle of the input layer and inference layer. We seek to combine Machine Learning and Deep Learning algorithms (LSTM), to consider most of the available information like the contextual information exploited by Bi-LSTM, which by itself does not require intense feature engineering.

Although we exploit information from the context with LSTM layer, the tagging decision in the inference layer is still local, so we do not use the neighbouring tagging decisions. Instead, the linear-chain CRF inference layer look for the best sequence of labels $y_1, \dots, y_m$ in all possible sequences, i.e. CRF get the maximum global score $C \in \mathbb{R}$ of the sentence given by the sum of transition scores and network scores, thus it learns the transition matrix T ($n \, x \, n$ labels) and vectors of scores of beginning and ending with a specific label (see Eq. 1) (Genthial, 2017), to capture linear dependencies (one step) between tagging decisions.

$$\begin{aligned} C(y_1, \dots, y_m) &= b[y_1] + \sum_{t=1}^{m} s_t[y_t] + \sum_{t=1}^{m-1} T[y_t, y_{t+1}] + e[y_m] \\ &= \text{begin} + \text{scores} + \text{transitions} + \text{end} \end{aligned} \tag{1}$$

For instance, the linear-chain CRF would choose the best score between all possible sequences of labels, for example the scoring for sentence *bleeding secondary_to warfarin* (see Fig. 3.1), the tagging of bleeding as an ADE should help to tag the next words with the correct labels. The sequence with the best score is ADE-None-Drug (score of chain 31), which is the correct prediction, meanwhile other algorithm that make independent predictions only based on the maximum score for each label, it would choose the sequence of labels ADE-ADE-Drug (score of chain 26).

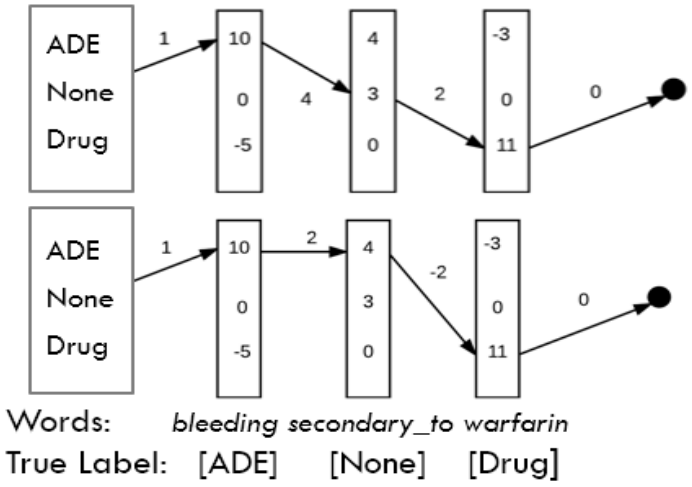


Figure 3.1: Sequence label scoring of sentence with linear-chain CRF

The full models are described in the following subsections.

3.3.2.1 Embedding Layer

The model implemented for the MADE dataset includes a specific word representation to exploit features of its entities, which extends the generic word embedding in French. Meanwhile, the model for the Spanish dataset includes generic word embeddings made with ordinary text in Spanish.

- **MADE Embedding**

We created a comprehensive word representation, which concatenates character-level representations, word embedding and POS features. The following subsections describe the word representation, as well as the full network using that representation to solve the NER task.

The character-level features can exploit prefix and suffix information about words (Lample, Ballesteros, Subramanian, Kawakami, & Dyer, 2016), to have closer representations among words of the same category. This is particularly useful for terms that may be Out-Of-Vocabulary (appearing in the test data and not in the training data). OOV is a common issue with domain specific words, and prefix and suffix representations can help a lot. For example, the words “Clonazepam” and “Lorazepam” both belong to the medication category in the medical context and may be OOV. However, they share the same suffix, making them closer to each other on a character-

level feature. Therefore we build a LSTM network (see sub-section 2.2) that get representations of words based on their characters.

The character-level embedding for words was built by a Bi-LSTM network (represented on the bottom left of Figure 3.2). First, each character takes an integer value from a lookup table, and then a one-hot vector replaces it. The final state of the forward and backward LSTM is the representation of the suffix and prefix of the word. The Character-level embedding is the concatenation of both LSTM layers, so with LSTM layers of 20 cells (units), we get a vector of 40 dimensions. This character-level representation is concatenated to the word embedding and the Part-of-speech feature to form the final comprehensive word representation (see Fig. 3.2) (Lample, Ballesteros, Subramanian, Kawakami, & Dyer, 2016).

Part-of-speech (POS) tags the words with labels like noun, verb, adjective, adverb, etc. It classifies words according to its roles within the grammatical structure of the sentence. Medications for example will always belong to the Noun category, making them close together with respect to this feature. The tagging was performed using an Averaged Perceptron algorithm (Honnibal, 2015).

Finally, we also use word embeddings learned from a large corpus, to consider the contexts in which words appear usually. It can create similar vectors (representations) for words that appear in similar contexts, such as the names of different countries. The word embedding (dimension 200) used was provided by (Yu, Jagannatha, Liu, & Liu, 2018), as well as another of 300 dimensions provided by FastText (Bojanowski, Grave, Joulin, & Mikolov, 2017). Both pre-trained with skip-gram using unlabeled data mainly from Wikipedia. The comprehensive word embedding is the input of a Bi-LSTM network, which takes a sequence of words and returns a sequence of hidden states at every time step (see Fig. 3.3).

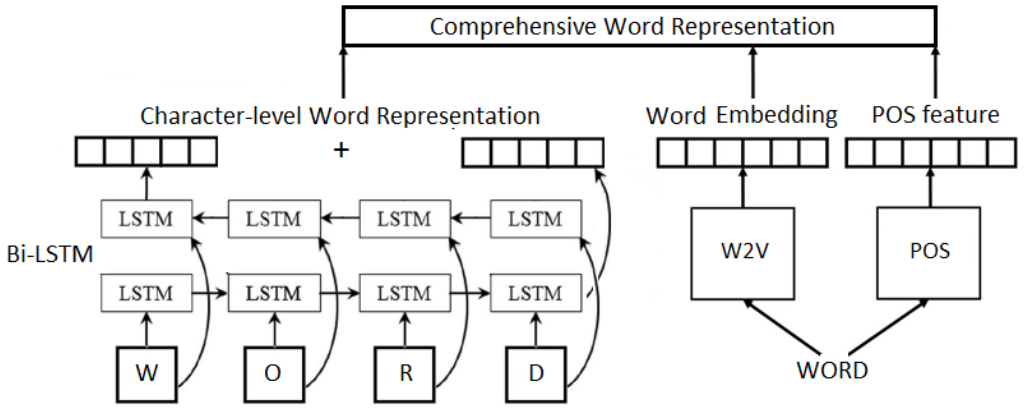


Figure 3.2: Comprehensive word representation

– **Spanish Embedding**

The NER model for clinical notes in Spanish used different word representations during the PharmacoNER challenge. We evaluated three word embeddings learned from different corpus. The embedding considers the contexts in which words appear usually, and then it can create similar representations (vectors) for words that appear in similar contexts, such as the names of different countries. We built a word embedding using Skip Gram algorithm and the training set, with size set at 300 dimensions, context window of size 5, and minimum word frequency of one to keep even the uncommon words such as underused medications. We used other Spanish word embedding created by FastText (Bojanowski, Grave, Joulin, & Mikolov, 2017), which was pre-trained on unlabeled Wikipedia data. Finally, we tested the model with the embedding learned during training of full layers of the neural network. Results were obtained on test set defined by the PharmacoNER challenge (Agirre, et al., 2019).

3.3.2.2 Neural Network Description

The input layer receives words represented by its corresponding vectors in the word embedding. Long Short-term Memory (LSTMs) Neural Networks can learn long term dependencies among the words of the sentence (Jagannatha & Yu, 2016). LSTM keeps information in a memory-cell (c_t gate) that is updated using input i_t and forget gates f_t (Lample, Ballesteros, Subramanian, Kawakami, & Dyer, 2016), and then it adjusts the output gate o_t and hidden state h_t (see equations 2, 3, 4 and 5). LSTM extracts contextual information to take into account long term dependencies among the words of the input sequence $x = (x_1, x_2, \dots, x_t)$ of length t (Hochreiter & Schmidhuber, 1997). LSTM keeps information through a memory cell (c_t), which is updated using input gate

i_t and forget gate f_t for every time step *t* (see Eq. 6), where $\otimes$ and σ are the element-wise product and sigmoid function respectively.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1}) \quad (2)$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1}\right) \quad (3)$$

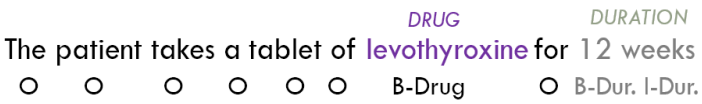
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1}) \quad (4)$$

$$h_t = o_t \otimes \tanh(c_t) \quad (5)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tanh(W_{xc}x_t + W_{hc}h_{t-1}) \quad (6)$$

The word embedding is the input of a Bi-LSTM network, which takes a sequence of words and returns a sequence of hidden states at every time step (see Fig. 3.4). A regular expression tokenizer pre-process the raw sentence into sequence of tokens. Sentences longer than the sequence length were cropped to size, and shorter sentences were pre-padded with masks. The forward and backward LSTM layers get hidden state sequences, which represent the left and right context of the sentence at every time step (word), and their concatenation is the representation of a word in context (Graves & Schmidhuber, 2005).

We implemented the BIO (Beginning-Inside-Outside tagging schema in order to manage entities with more than one word (Ramshaw & Marcus, 1999), the first word received the label market as Beginning (B) and the remaining words the same label market as Inside (I) words of the full entity. Thus we have in training double types of labels plus the None (O) label, then we got 21 labels for MADE dataset instead the eleven original labels (ten categories plus None) in the inference layer (last layer in Figure 3.3), which are reshape to the original labels during the post-processing, for example:



We use Dropout (at 0.5) to prevent over-fitting as a regularization method for the network. The word embedding size provided by FastText is 300 (pre-trained with unlabelled data from Wikipedia) (Bojanowski, Grave, Joulin, & Mikolov, 2017), and the best sequence length was 70 words for MADE dataset.

Layer (type)	Output Shape	Parameters
embedding_1 (FastText)	(None, 70, 300)	3934500
dropout_1 (Dropout)	(None, 70, 300)	0
bidirectional_1 (Bi-LSTM)	(None, 70, 200)	320800
Dense_1 (Dense Layer)	(None, 70, 21)	4221

Total params: 4,259,521
Trainable params: 325,021
Non-trainable params: 3,934,500

Figure 3.3: Baseline model based on LSTM for NER on MADE dataset

The bidirectional LSTM provides scores for every possible label for each word, its output (hidden states) feed the inference layer for tagging each word independently. However it does not take into account the correlations between adjacent labels that can help in sequence labelling problems (Lample, Ballesteros, Subramanian, Kawakami, & Dyer, 2016). Therefore, we put Conditional Random Fields (CRF) as inference layer in the final model (see Fig. 3.5) instead the Dense layer (fully connected layer) of the baseline model (see Fig. 3.3), which connects the LSTM hidden states to each possible label. CRF is a probabilistic model that have been used for sequence labeling tasks due to their ability to model the dependencies in the outputs of a sequence (Lafferty, McCallum, & Pereira, 2001), then we minimize the error in the prediction of a chain of labels, not just every label independently. For example, in the sentence "*the patient has internal bleeding* (Adverse Event) *secondary to warfarin* (Medication)", the label for Adverse Event entity is strongly related to the Medication label, then *Warfarin* is labelled as Medication using information of previous annotation (*internal bleeding*), which is exploited by CRF. Thus, we have a combination of LSTM and CRF models (BiLSTM-CRF) for Named Entity Recognition (see Fig. 3.4).

At the end of the last layer, the Softmax function (over the score of all possible labels) normalizes the probability for each label, so the final output are values between 0 and 1 that together sum 1, which is used to get the label for each word. The prediction is the label with the maximum probability of Softmax, which is evaluated with the correct class (true label). The target labels consist in an integer vector where each element represents the position of the number 1 in a one-hot encoding. Categorical

cross-entropy is the loss function to calculate the error (cost) during the training, which penalizes the deviation between the predicted and target (true) labels. Then, the optimization function will minimize the loss of the correct labels sequence.

The input and output of the neural network will be a sequence of words embedding and its corresponding labels for training (see Fig. 3.4), and the neural network will try to learn a model that minimize the error of label prediction. The implementation was made through Keras Python library with Tensorflow-GPU background, for parallel execution on computing cluster nodes with GPUs.

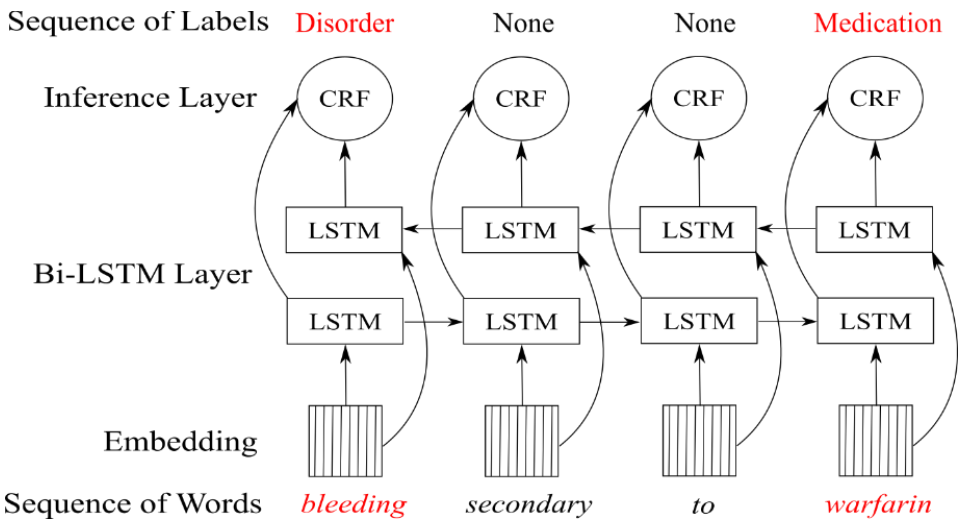


Figure 3.4: Final model for sequence tagging

Layer (type)	Output Shape	Parameters
embedding_1 (FastText)	(None, 70, 300)	3934500
dropout_1 (Dropout)	(None, 70, 300)	0
bidirectional_1 (Bi-LSTM)	(None, 70, 200)	320800
crf_1 (CRF Layer)	(None, 70, 21)	4704

Total params: 4,260,004
Trainable params: 325,504
Non-trainable params: 3,934,500

Figure 3.5: Model based on LSTM and CRF for NER on MADE dataset

3.3.2.3 Other variations of the model

The attentional model was implemented as extension of our model. First, we take the hidden output h_t of LSTM as input of the Attention layer (see Eq. 7) (Zhou, et al., 2016), to calculate the score of how much attention should be put on the *i-th* hidden state, these scores are normalized by Softmax function to create another vector (see Eq. 8), where *t* is the sentence length or number of time steps and w and w^T are a trained parameter vector and its transpose. Then, a Context vector c_t of the sentence is formed by a weighted sum of these output vectors (see Eq. 9). Such as in the Sequence to Sequence problem by (Luong, Pham, & Manning, 2015), we concatenate the output of LSTM and context vector (see Eq. 10 and Fig. 3.7). The concatenation becomes the new hidden state h'_t (final word representation) used for classification (see Fig. 3.6). Recently studies tried to include Attentional models and CRF models in the same network (Luo, et al., 2018), with no significant performance improvement.

$$M = \tanh(h_t) \tag{7}$$

$$\alpha = softmax(w^T M) \tag{8}$$

$$c_t = \alpha^T h_t \tag{9}$$

$$h'_t = \tanh(w_c[c_t; h_t]) \tag{10}$$

Layer (type)	Output Shape	Parameters	Connected_to
input_1 (InputLayer)	(None, 70)	0	
embedding_2 (FastText)	(None, 70, 300)	3934500	input_1
dropout_2 (Dropout)	(None, 70, 300)	0	embedding_2
bidirectional_2 (Bi-LSTM)	(None, 70, 200)	320800	dropout_2
dense_1 (Dense)	(None, 70, 1)	201	bidirectional_2
flatten_1 (Flatten)	(None, 70)	0	dense_1
activation_1 (Activation)	(None, 70)	0	flatten_1
repeat_vector_1 (RepeatVector)	(None, 200, 70)	0	activation_1
permute_1 (Permute)	(None, 70, 200)	0	repeat_vector_1
multiply_1 (Multiply)	(None, 70, 200)	0	bidirectional_2 permute_1
lambda_1 (Lambda)	(None, 200)	0	multiply_1
repeat_vector_2 (RepeatVector)	(None, 70, 200)	0	lambda_1
concatenate_2 (Concatenate)	(None, 70, 400)	0	bidirectional_2 repeat_vector_2
dense_2 (Dense)	(None, 70, 200)	80200	concatenate_2
dense_3 (Dense)	(None, 70, 21)	4221	dense_2

Total parameters: 4,339,922
Trainable parameters: 405,422
Non-trainable parameters: 3,934,500

Figure 3.6: Model based on LSTM and Attentional model for NER on MADE dataset

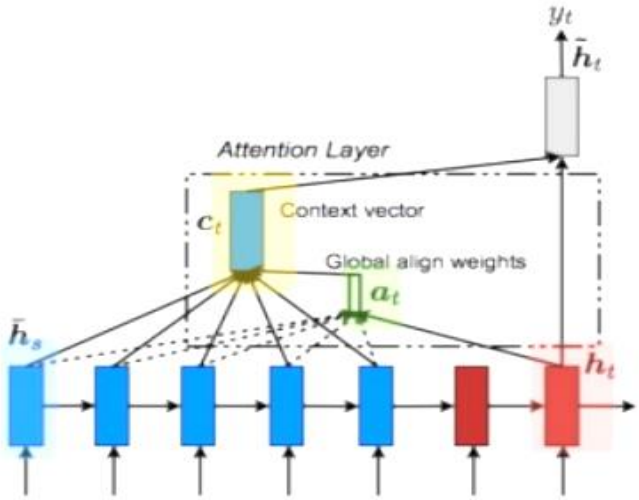


Figure 3.7: Attentional model (Luong, Pham, & Manning, 2015)

3.4 Results and Discussion

We show the results for each dataset presented in Section 3.4.2.1, which were validated by the organizations of every challenge.

3.4.1 MADE Results

The models were compared with the same parameters and training dataset as those of the MADE challenge. We split the training dataset into 20% and 80% for training and development set, respectively. We trained different models on randomly shuffled clinical notes (876 documents split between training and development set) of MADE dataset (Jagannatha & Yu, 2016). Then, the models were tested on test dataset of 213 clinical notes established by the MADE challenge. We calculated the mean precision, recall, and F1 measure for all type of relations (see Table 3.4).

The results are shown in Table 3.2, with results for models without pre-trained word vectors (baseline), models using a pre-trained 200-dimensional embedding W2V (Jagannatha & Yu, 2016; Yu, Jagannatha, Liu, & Liu, 2018), and the last model with pre-trained W2V(FT) with FastText of 300 dimensions (Bojanowski, Grave, Joulin, & Mikolov, 2017), POS features (46 tags) and Character-level word representation Char(LSTM) of length 40. First, we set up same hyper-parameters for fair comparison between all models with batch size of 32 sequences, sequence length of 60, 100 LSTM

nodes (x2 bidirectional hidden layer) and initial learning rate 0.1 using Adagrad optimizer (see Table 3.2). Finally, we look for the optimal set up of hyper-parameters for our best models in MADE challenge, by adjusting the hyper-parameters values during different runs until find the maximum accuracy (see Table 3.3).

Model	Recall	Precision	F1
Baseline (LSTM)	0,686	0,704	0,695
+ W2V(1)	0,668	0,689	0,678
+ Char(LSTM) + POS	0,659	0,678	0,668
+ W2V(FT)	0,694	0,721	0,707
+ W2V(FT) + POS	0,691	0,719	0,704
+ W2V(FT) + Char(LSTM)	0,692	0,724	0,708
+ W2V(FT) + Char(LSTM) + POS	0,700	0,721	0,710

Table 3.2: Performance of models with MADE dataset

Note: Hyper-parameters batch size 32, sequence length 60, 100 LSTM cells, learning rate 0.1 (Adagrad optimizer).

We improved more the performance using the largest word embedding of FastText (W2V(FT)) than using word2vecembedding W2V(1) (Jagannatha & Yu, 2016) pre-trained with Skip-gram algorithm. The model with FastText W2V(FT) obtained about 4.1% more than W2V(1) in F1. We observed the highest improvement over the baseline model (LSTM) with all the features together, i.e. the LSTM model with W2V(FT)+Char(LSTM)+POS, it increases the F1 about 2.1%. Models including W2V(FT) only with the Char(LSTM) provided small increase in F1, while POS alone does not increase anything (see Table 3.2).

The best model (LSTM + W2V+Char(LSTM)+POS) was trained during MADE challenge using all training files, then it was evaluated in the prediction of annotations for test dataset established by the MADE Challenge (Yu, Jagannatha, Liu, & Liu, 2018). Table 3.3 shows the official results validated by the MADE challenge, the best result of 2 runs for standard with W2V(1) and extended evaluation (with W2V(FT)). The usage of more hidden units (200 or 300 LSTM cells) did not significantly influence the model performance, and big values (60, 70, 80) of the sequence length (number of words by sequence) provided better results in our experiments with the clinical notes of MADE dataset. The most appropriate initial value for the learning rate was 0.1 (using Adagrad), a smaller learning rate decreased the performance and increased the running time.

Model	Recall	Prec.	F1
LSTM + W2V(1)+Char(LSTM)+POS	0,720	0,681	0,700
LSTM + W2V(FT)+Char(LSTM)+POS	0,748	0,716	0,732

Table 3.3: Performances of models for NER task in MADE Challenge (test set)

We obtained low accuracy with the best model (in test dataset) for some categories (see Table 3.4), mainly for ADE and Duration categories. Their performance is lower than other categories mostly because the training dataset have an imbalance problem, i.e. very low and high number of samples in some categories. ADE and Duration only have about 2.2% and 1.1% of the total number of entities respectively, otherwise SSLIF has about 50% of the total number of entities.

Entity Type	Recall	Precision	F1	Annotations (Training)	% total Ann.
Drug	0,8079	0,8724	0,8389	13507	20,0
Indication	0,5031	0,5079	0,5055	3168	4,7
Frequency	0,7071	0,6384	0,6710	4148	6,1
Severity	0,6929	0,6852	0,6890	3374	5,0
Dose	0,8052	0,7752	0,7900	4893	7,2
Duration	0,4511	0,4196	0,4348	765	1,1
Route	0,8380	0,8763	0,8568	2278	3,4
ADE	0,3457	0,5560	0,4263	1509	2,2
SSLIF	0,7866	0,6896	0,7349	34056	50,3

Table 3.4: Performance by category on test dataset of our best model in MADE challenge

ADE entities are mostly confused with SSLIF (see Table 3.5), it got 53% of total mistakes (758) with SSLIF, due to the common vocabulary between ADE and SSLIF entities, e.g. words like fever could be and ADE or SSLIF according to the sentence, meanwhile the other categories have the highest confusion only with None category. There are also 36% of ADE entities confused as the None category, and most of the remaining ADE entities are confused only with Indication (8%). SSLIF is a vague category that create high confusion also with Indication and None categories (see Table 3.5), because it has a common vocabulary mixed with *Sign, Symptom and another Disease, ADE or Indication*, category also called SSD.

Entity Type	ADE	Dos.	Drug	Dur.	Freq.	Ind.	None	Rou.	SSL-IF	Sev.
ADE	1461		9			61	279	1	404	4
Dosage	2	11097	66	2	57		517	29		
Drug	2	97	16256	2	6	12	768	7	15	
Duration		15	1	1293	18		207			
Frequency		49	4	15	10909		591	14		
Indication	68		19			4770	419		1254	10
None	136	412	505	127	563	376	884608	5	6296	443
Route		30	17		24		188	3660	1	
SSLIF	111		15			424	5732	2	60724	95
Severity	3	3	1			3	671		131	4226

Table 3.5: Confusion matrix between entities

Table 3.6 shows results obtained with the updated version of our model after the MADE challenge, based on the combination of Machine Learning and Deep Learning algorithms for NER task. The results show that a combination of LSTM and CRF models (LSTM+CRF) is effective to get better performance than the baseline (LSTM). CRF layer contributes to outperform considerably (+12.3%) our best result in MADE Challenge, and the best model reached state-of-the-art leaders (see Table 3.8).

Contrary to the results with LSTM-based models, LSTM+CRF models did not get more accuracy using a wide character representation or POS tagging. We also researched algorithm variations with Attentional layer, LSTM+Att+W2V(FT) and LSTM+Att+CRF+W2V(FT) models could not yield more performance than LSTM+CRF+W2V(FT), which reiterates the importance of CRF for the inference layer. We observe in Table 3.7 the same performance patterns for the categories than Table 3.4, with the highest Precision for Drug entity and Route category getting the highest F1 and Recall.

Model	Recall	Precision	F1
LSTM (Baseline)	0,686	0,704	0,695
+ W2V(1)	0,668	0,689	0,678
+ W2V(FT)	0,694	0,721	0,707
+ W2V(1) + Char(LSTM) + POS	0,720	0,681	0,700
+ W2V(FT) + Char(LSTM) + POS	0,748	0,716	0,732
+CRF +W2V(FT)	0,773	0,804	0,788
+CRF +W2V(FT)*	0,834	0,813	0,823
+CRF +W2V(FT)+ Char(LSTM)*	0,826	0,806	0,816
+CRF +W2V(FT) + POS*	0,832	0,805	0,818
+Att +W2V(FT)*	0,802	0,760	0,781

Table 3.6: Performance of models

Note:*Models that included BIO tagging schema

Entity Category	Recall	Precision	F1-score	Annotations (Training)	% total Ann.
Drug	0,906	0,901	0,903	13507	20,0
Indication	0,673	0,656	0,665	3168	4,7
Frequency	0,853	0,787	0,819	4148	6,1
Severity	0,848	0,788	0,817	3374	5,0
Dose	0,835	0,833	0,834	4893	7,2
Duration	0,752	0,629	0,685	765	1,1
Route	0,933	0,888	0,910	2278	3,4
ADE	0,497	0,735	0,593	1509	2,2
SSLIF	0,839	0,800	0,819	34056	50,3
Overall	0,834	0,813	0,823	67698 total	100%

Table 3.7: Performance by category on test dataset of our best model

We made the comparison of model with actual algorithms of state-of-the-art presented in MADE challenge (see Table 3.8). We got 0.829 F1 compared to about 0.82 of top three of the teams ranking for Standard Evaluation based on strict matching of NER task (Jagannatha, Liu, Liu, & Yu, 2019), using only standard resources, i.e. MADE resources such as released training data or pre-trained word embedding. On the other hand, our updated model (LSTM+CRF) got similar performance than the top models presented in MADE challenge. There the IBM Research team address the OOV problem using specific embedding (for medical knowledge) trained on clinical notes (EHR), they trained a multi-layer neural network to learn a mapping function, which maps initial embeddings to updated embeddings for the words that appear in training data. An additional strategy is still necessary to overcome the local optima solution

found by the models, such as an additional layer that could work as feature extractor, located just after the embedding layer.

Model	Team	F1
W2V+Char(LSTM)+ LSTM+CRF	Worcester Polytechnic Institute	0.829
W2V+Char(LSTM)+POS+ LSTM+CRF	IBM Research	0,829
W2V+Char(LSTM)+ LSTM+CRF	University of Florida	0,823
W2V(FT)+Char(LSTM)+POS+ LSTM	Our model	0,700
W2V(FT)+LSTM+CRF	Our model updated after challenge	0,823

Table 3.8: NER task results in MADE Challenge (Strict Evaluation)

3.4.2 Results with Dataset in Spanish

The models have been compared with the same hyper-parameters and datasets distribution established by PharmaCoNER Challenge. The Train set is composed of 500 clinical cases and Development set is composed of 250 clinical cases. Test set (only text files) is composed of 3751 clinical cases, including an additional collection of documents (background set) to make sure that participating teams will not be able to do manual corrections and also that these systems are able to scale to larger data collections (Agirre, et al., 2019). Then the Test set with Gold Standard annotations consists of 250 clinical cases.

The results belong to three models (see Table 3.9), first, a model with embedding learned during the training of all layers (named W2V(learnt)), other model with embedding pre-trained using Skip Gram and training set (named W2V(pre-trained)), and the last model used the pre-trained FastText embedding of 300 dimensions (W2V(FT)) (Bojanowski, Grave, Joulin, & Mikolov, 2017). We report performance metrics (Precision, Recall and F1) for each model, but the results analysis is centred on F1 score (average of all classes) because it combines precision and recall. Results are based on Exact Match metric, which considers a term (word or group of words that have a label) as correctly classified only if all the words in the term received the correct label.

The models were created with all training texts, and then the model predicted the annotations for the test dataset defined by the challenge organization. Table 3.9 shows the results for one run performed during the PharmaCoNER challenge, results published in (Agirre, et al., 2019), where we set same hyper-parameters for equal comparison

between models, with batch size 32, sequence length 50, 100 LSTM cells and initial learning rate 0.001 (Adagrad).

Model	**Recall**	**Prec.**	**F1**
LSTM+CRF+W2V(learnt)	0.6908	0.8465	0.7608
LSTM+CRF+W2V(pre-trained)	0.1493	0.6335	0.2416
LSTM+CRF+W2V(FT)	0.6892	0.8066	0.7433

Table 3.9: Overall performance for NER task on test set

We obtained the best performance with the embedding learned during training (model with W2V(learnt)) than the word embedding of FastText W2V(FT). The model with W2V(learnt) achieved about 2.4% more F1 than the model with W2V(FT) embedding. We obtained null performance of our best model with test dataset for *No_Normalizables* category (see Table 3.10), which has only 10 true annotations in the test set, all of them predicted as False Negative, meanwhile the next category with low F1, *Unclear* Entity type has 34 true annotations (three times more annotations), for what we obtained 13 False Negatives. The performance for *No_Normalizables* is lower than other categories because the training dataset have an imbalanced distribution of annotations, i.e. low number of samples in some categories and high number in the other categories (*Normalizables* and *Proteinas*). The dataset contains only 0.8% of total number of annotations for *No_Normalizables* category, otherwise *Normalizables* category has about 64% of the annotations. Table 3.11 shows the highest precision and F1 for *Normalizables* category, and the highest recall for *Proteinas* category, both are the categories with more annotations.

Entity Type	LSTM+CRF +W2V(learnt)	LSTM+CRF +W2V(pre-trained)	LSTM+CRF +W2V(FT)	**Anno-tations**	**%total Ann.**
Normalizables	0.7795	0.2862	0.7684	4426	63,9
No_Normalizables	0	0	0	55	0,8
Proteinas	0.7531	0	0.7333	2291	33,1
Unclear	0.7241	0.1998	0.7238	159	2,3
Overall	0.7608	0.2416	0.7433	6931	100

Table 3.10: F1 score by category on test dataset

Entity Type	Recall	Precision	F1-score	Anno-tations	%total Ann.
Normalizables	0.6886	0.8981	0.7795	4426	63,9
No_Normalizables	0	0	0	55	0,8
Proteinas	0.7101	0.8016	0.7531	2291	33,1
Unclear	0.6176	0.8750	0.7241	159	2,3
Overall	0.6934	0.8497	0.7608	6931	100

Table 3.11: Performance by category on test dataset of the best model

3.4.3 Important issues in clinical notes

We see high influence of the embedding layer and tokenization for NER in medical data. It is decisive for improving performance an appropriate word representation and text tokenization specialized in medical entities, besides the provision of more informative input features.

We found complex named entities in the dataset in Spanish clinical dataset, especially protein entities such as “CAM5.2” and “S-100”. Then, we need a dedicated tokenizer to avoid the split of these named entities. We also can use Piece2Vec tokenizer, which is able to reduce the Out-of-Vocabulary (OOV) problem because it represents the unknown words with vectors of common pieces of words.

OOV is a significant problem in medical corpus like our datasets, mostly because entities involved in relations are medications, proteins or chemical names, OOV in standard embedding, for example entities “Tc99m-MDP” and “6-Metil-Prednisolona” that belong to *Normalizables* category. Then a standard embedding is not enough for medical corpus, it is necessary a dedicated embedding trained with medical corpus and target language available (list of events and drugs) in order to minimize the number of words without vector representation (OOV), the model needs to learn the specific vocabulary such as protein and chemical names provided in specialized dictionaries.

We can add other features such as suffix and prefix components to provide a vector representation for words without representation, such as medication and chemical names usually unknown for standard embeddings. It would provide a part of the wide vector representation, which is composed by several levels for word, characters and other representations.

3.5 Conclusion

We built an appropriate model to recognize medical entities on clinical notes, we studied the model in different datasets mainly focus on ADE and Medication entities. The model requires good input features for training, so we built character-level features extracted with another LSTM, that were used in conjunction with word representations as a comprehensive word representation. This conjunction of features increased the performance of LSTM, and models using FastText embedding obtained better results than embeddings trained with word2vec embedding (Skip-gram algorithm). However, it does not allow to LSTM model (alone) to reach the best performance achieved for the task, so we did an extension of the model with a CRF layer, because it considers the dependency between chains of successive labels in the inference layer, which is ignored by models based only on LSTM. However we got low accuracy for the ADE label, then we should extend the model through Transfer Learning, for example, inserting another layer with a pre-trained model such as BERT for feature extraction (Devlin, Chang, Lee, & Toutanova, 2018).

We also work with clinical notes in another language different to English (challenge in Spanish), we tried different word representations to increase the performance of our best model (LSTM+CRF). The embedding of the best model was learned during training, probably due to the Out-of-Vocabulary (OOV) problem in the pre-trained embeddings, which do not have word representation for entities such as proteins presented in the test set. Therefore, we suggest to create a dedicated embedding for clinical notes in Spanish, in order to reduce Out-of-Vocabulary problem through a more suitable tokenizer, based on Piece2Vec tokenizer (used by BERT) that splits unknown words in word pieces that have a vector representation, or specific vocabulary for protein and chemical names added during the training of an embedding.

The NER model is the first stage in our full approach for Adverse Drug Reaction detection. The next chapter explains the Relation Extraction task with a supervised approach, which takes as input the entities identified by the NER model.

CHAPTER 4

Relation Extraction in Clinical Notes

4.1 Introduction

The Information Extraction of medical events from clinical notes of EHR (Electronic Health Records) is relevant for post-marketing surveillance in Pharmacovigilance (Drug Safety). Since clinical records contain enough information about patient health than structured documents, this is useful to detect side effects of medications and to improve drug safety. Patients are often subject to multiple treatments, which may be the cause of adverse side effects, formally known as Adverse Drug Event (ADE). ADE refers to any adverse event occurring at the time a drug is used, whether it is identified as a cause or not. Therefore, it is necessary to establish whether there are relations between medications and ADEs mentions in clinical notes, which is a Relation Extraction task. If a relation between an ADE and drug is detected, then it is considered as an Adverse Drug Reaction (ADR).

The dataset released for the MADE challenge provides clinical notes with annotations for Relation Extraction task (Yu, Jagannatha, Liu, & Liu, 2018), which also works in the first level for Named Entity Recognition task. The annotations are mentions of medical entities like medications, ADE, and indications. The relation between the entities is identified and classified using the annotated relations that are also provided in the dataset for supervised learning.

Recently, models based on Machine Learning and Deep Learning improved the performance for detection of relations between medical entities (Jagannatha, Liu, Liu, & Yu, 2019). This work proposes an enhanced model of Deep Learning with additional external features for Relation Extraction in clinical notes.

4.2 Related Work

Approaches for relation extraction can be classified into rule-based, lexicon-based, and supervised learning (Jagannatha, Liu, Liu, & Yu, 2019). Nowadays, works are more

focused on supervised learning due to the high performance of Machine Learning and Deep Learning methods. These approaches have been applied in general domains, using named entities like person and organization. A named entity is a term (composed by one or more words) that belong to any defined category. In this work, ADR extraction is based on plain clinical notes with annotations for relations and named entities, provided in the MADE dataset (Jagannatha & Yu, 2016). MADE is the first high-quality dataset for ADR research (Jagannatha, Liu, Liu, & Yu, 2019). Other datasets like i2b2 do not provide annotations for ADR relations (Uzuner, Solti, & Cadag, 2010), which only include relations of medical problems, tests, and treatments (Li, et al., 2013). Thus, it is not very useful for our Pharmacovigilance research field. The MADE dataset has several relation types between two different entities (see Fig. 2), which can occur within a sentence or across multiple sentences in a note. For instance, ADE–Drug pair conforms de "Adverse" relation, where ADE is an adverse effect of the Drug prescribed, in the SSD–Severity pair, Severity entity is an attribute of SSD (Sign, Symptom and another Disease, ADE or Indication).

The Relation Extraction problem can be solved based on the information extraction of the data between candidate entities. The classification methods are based on Deep Learning models such as Bidirectional Long Short-Term Memory (LSTM) with Attention layer (Dandala, Joopudi, & Devarakonda, 2018), and Machine Learning algorithms such as Random Forests (Chapman A. B., Peterson, Alba, DuVall, & Patterson, 2018; Magge, Scotch, & Gonzalez-Hernandez, 2018) and Support Vector Machines (SVM) (Xu, Yadav, & Bethard, 2018). SVM uses maximum margin loss to train the classifier, and Random Forest uses the combined score from a collection of decision trees to produce the class prediction (Jagannatha, Liu, Liu, & Yu, 2019). The Machine Learning works mentioned before were implemented using Scikit-learn python package (Chapman A. B., Peterson, Alba, DuVall, & Patterson, 2018; Magge, Scotch, & Gonzalez-Hernandez, 2018; Xu, Yadav, & Bethard, 2018). The classification is divided into two separate classification procedures to improve the accuracy by (Chapman A. B., Peterson, Alba, DuVall, & Patterson, 2018), using the absence of a relation between entities as another class. First, the binary classification procedure predicts if there is a relation between two entities, to remove all the pairs with no relations. Then, the multiple classification procedure predicts the relation type for the remaining pairs, i.e., all candidate pairs that were predicted to have a relation.

Recently, approaches based on LSTM neural networks have been proposed in (Munkhdalai, Liu, & Yu, 2018) and (Dandala, Joopudi, & Devarakonda, 2019). The researchers in (Munkhdalai, Liu, & Yu, 2018) take the previous words with a fixed window size of both candidate entities as an input of the LSTM layer. In the LSTM model proposed by (Dandala, Joopudi, & Devarakonda, 2019), the input of the LSTM layer is the sentences between the entities of the relation, included the sentences in which the entities appeared. This network also takes entity types (Named entity labels) and positional indicators around the source and target concepts as inputs. It includes external knowledge for a medical relation, which is an ensemble association scoring between Drug–SSD pair. They calculate the strength of association using two distinct systems (ensemble system), which takes as input the CUIs (Concept Unique Identifiers) sets for SSD and medications of the Unified Medical Language System (UMLS) (Bodenreider, 2004) provided by a UMLS CUI finder. The scores were additional input to the Attention-LSTM model, added before the connection with the dense layer, see Figure 3 in (Dandala, Joopudi, & Devarakonda, 2019).

All of these works were presented during the MADE Challenge (Yu, Jagannatha, Liu, & Liu, 2018) (see Table 4.1), which is a good benchmark because the algorithms were executed in the same conditions (rules). Results of other previous works cannot be compared directly since they used different biomedical text datasets, or they only extracted relations within a sentence, instead of any number of sentences.

Model	**Recall**	**Prec.**	**F1-score**
Random Forest (Chapman A. B., Peterson, Alba, DuVall, & Patterson, 2018)	0.881	0.857	**0.868**
Attention LSTM (Dandala, Joopudi, & Devarakonda, 2019)	0.874	0.809	0.840
SVM (Xu, Yadav, & Bethard, 2018)	**0.885**	0.785	0.832
Random Forest (Magge, Scotch, & Gonzalez-Hernandez, 2018)	0.770	**0.869**	0.816

Table 4.1. Relation Extraction results in MADE Challenge, NER Task (Jagannatha, Liu, Liu, & Yu, 2019)

Systems based on Random Forest archives the best result (Chapman A. B., Peterson, Alba, DuVall, & Patterson, 2018). However, results of an LSTM-based model reported after the MADE Challenge (Dandala, Joopudi, & Devarakonda, 2019), an updated version of the system presented by (Dandala, Joopudi, & Devarakonda, 2018), outperformed this system with 0.872 F1 (see Table 4.3), indicating the effectiveness of

the Deep Learning models. Therefore, we use LSTM-based networks for relation extraction, with LSTM alone as baseline system.

4.3 Relation Extraction Model

The Relation Extraction task is represented as a supervised classification problem, i.e., thus training is performed given the named entity annotations and the relation annotation (target label), and the model trained can predict the relations between any possible entity pair (see Fig. 4.1). It's a pairwise classification problem across the defined type of relations, plus one class for pairs with no relations. The entities can participate in one or many relations or do not participate in any relation. There are two phases before the supervised training, which are Candidate Generation and Feature Extraction of relations.

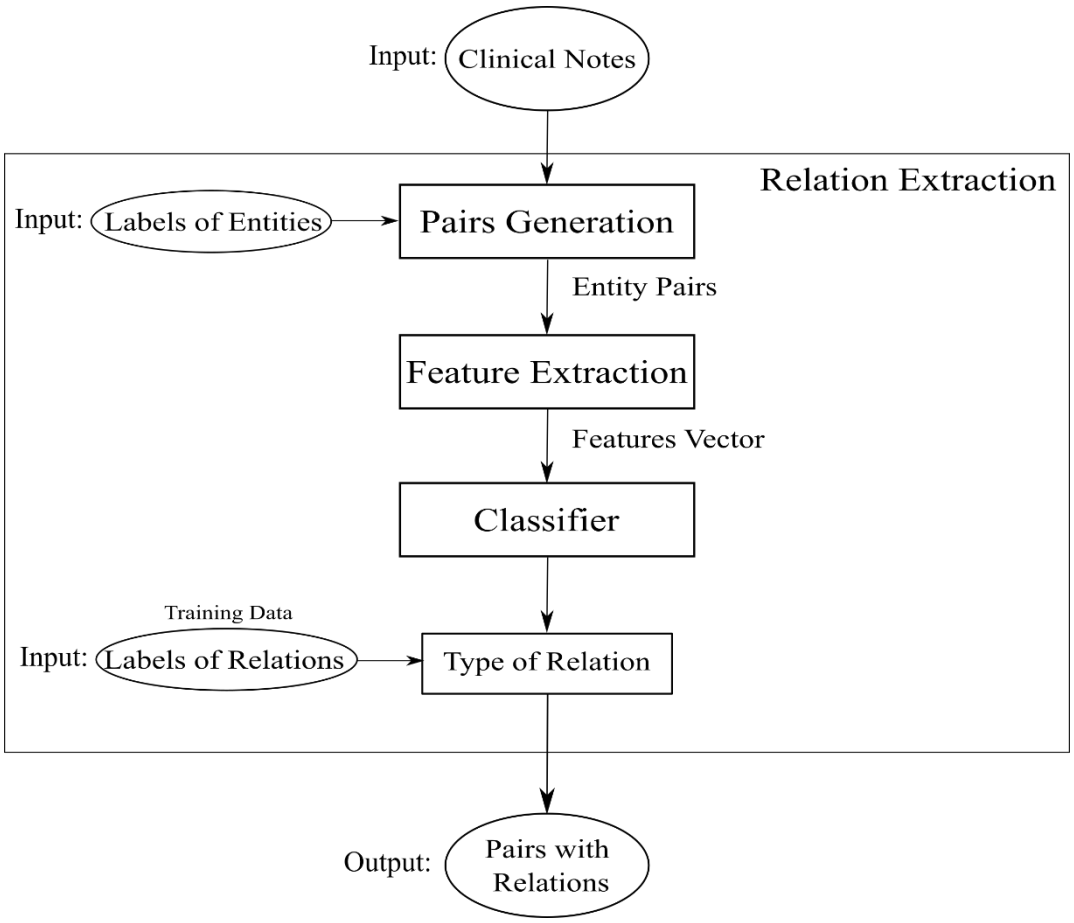


Figure 4.1: Relation Extraction module

4.3.1 Candidate Generation

The entities that participate in a relationship can appear anywhere in a clinical note in different sentence or paragraph. Then, if all possible entity pairs are created, 100% of recall would be obtained. However, many negative relations are obtained, which is much higher than the positive relations. It implies an unbalanced dataset, with a training procedure of high computational cost. Therefore, the negative samples are under-sampled randomly at the end of the candidate generation, such as was done in (Quirk & Poon, 2016; Peng, Poon, Quirk, Toutanova, & Yih, 2017) to balance the dataset. We create candidate pairs of medical entities that may have a relation, according to the following rules:

The maximum number of sentences (distance) allowed between the entities of the candidate pair. If it is high enough, we would create almost 100% of the positive pairs and cause the imbalance problem as mentioned above. Then, we control the number of negative examples using this variable of distance.

The type of entities that can participate in a candidate pair is restricted to the defined relations by the dataset. We do not allow incoherent negative relations like Duration-Dosage, but we also experiment allowing all the possible combinations. Some authors removed the entity pairs that have other types of labels (Chapman A. B., Peterson, Alba, DuVall, & Patterson, 2018; Chapman A. B., Peterson, Alba, DuVall, & Patterson, 2019).

Finally, we make a random sampling of negative relations to get an appropriate proportion regarding positive samples, like the same number or the double of positive relations, in that way we reduce many negative examples. In (Chapman A. B., Peterson, Alba, DuVall, & Patterson, 2018) the authors sampled as many negative instances as the number of entity pairs with similar types, in other works were sampled approximately the same number of negative examples as positive ones (Quirk & Poon, 2016).

4.3.2 Feature Extraction

We extracted the following features proposed in different works (Chapman A. B., Peterson, Alba, DuVall, & Patterson, 2018; Xu, Yadav, & Bethard, 2018; Swampillai & Stevenson, 2011), for each candidate pair to train the relation classifier:

-Information about Candidate Entities: entity types and words of the entities being considered for a relation.

-Information about Entities Between: number of entities (named entity annotations) and Entity types located between the candidate entities.

-Distances: number of words and sentences between the entity pair. We also can reinforce this important feature with another variable to inform whether both candidate entities are in the same sentence.

-Sequential information: all words (text) between the candidate entities (included), which are the logical units of the sequential input (contextual information) of LSTM layer (Hochreiter & Schmidhuber, 1997).

4.3.3 Training

The base structure of our model consists of two layers of neural networks, Bi-LSTM (Bidirectional LSTM) layer and Dense layer. We seek to combine feature-based approaches (knowledge provided by feature engineering) and Deep Learning approaches (LSTM), to consider most of the available information like the contextual information exploited by Bi-LSTM, which by itself does not require intense feature engineering.

The input of the bidirectional LSTM layer is a sequence of word embedding for each relation, with all the words between the candidate entities (included), provided by the embedding layer of pre-trained W2V (Bojanowski, Grave, Joulin, & Mikolov, 2017). We only take the last hidden state (h_t) of LSTM for all time steps of the sequence (length *t*), i.e., the concatenation of forward and backward LSTM hidden output, which represents the contextual information of the relation. LSTMs extract contextual information to take into account long term dependencies among the words between the two entities that conforms a relation.

We include the external features as an additional input to the dense layer, which is a vector of all the available features. This features vector is concatenated to the Bi-LSTM output (last hidden state) just before the connection with the dense layer (see Appendix A). The dense layer (last layer) is connected to the vector of possible labels (see Fig. 4.2), to get the probability score (through Softmax) for each type of relation, and the

relation label with maximum score is the final output of the model (or the target label in training). Cross Entropy loss is implemented to calculate the prediction error during the training of the relation classifier.

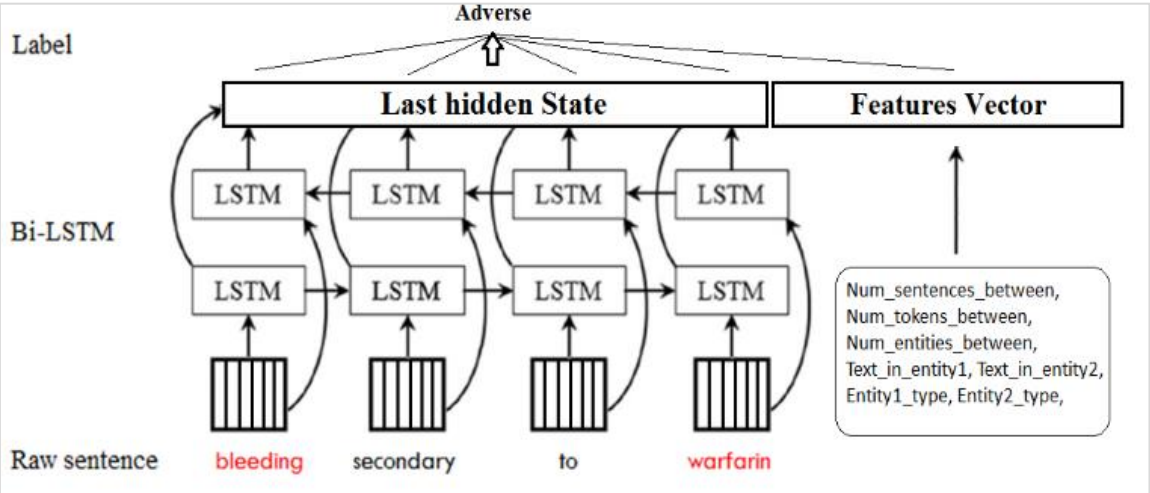


Figure 4.2: Relation Extraction model

We represent the features with one-hot encoding for categorical features (text in annotation 1, first entity type, etc), and numerical for the others features (number of sentences between candidate entities, number of entities between candidate entities, number of tokens between candidate entities), like in the following example of feature vector for an Adverse relation (ADE-Drug pair): “The patient has <ADE> bleeding </ADE> secondary to <DRUG> warfarin </DRUG>” (see Appendix B). We extracted the features:

num_sentences_between: 1, num_entities_between: 0, text_in_anno1: “bleeding”, second_entity_type: <DRUG>, text_in_anno2: “warfarin”, first_entity_type: <ADE>, entities_between: < >, num_tokens_between: 2

4.3.4 Transfer Learning

In Transfer learning we can use the knowledge gained while solving one problem (stored in pre-trained model) to solve other related problems. Recently, this field is dominated by a language representation model called BERT (Bidirectional Encoder Representations from Transformers) developed by Google (Devlin, Chang, Lee, & Toutanova, 2018). The pre-trained BERT model can be fine-tuned with additional

output layer to create models for many tasks of natural language processing, such as question answering (Q&A) and NER tasks.

We make the fine tuning of BERT version for Sentence Pair classification tasks (see Fig. 4.3), which gets two sequences as input (the question and its corresponding answer in Q&A task). We adapted the model to our Relation Extraction task, then we put the text inside both entities of the relation as input, and the label of the relation as target, in that way the trained model can create a vector representation for each candidate pair, in order to predict independently the type of relation between de entities, see results in Table 4.3. We also could include features of each candidate pair (or relation) as another embedding level.

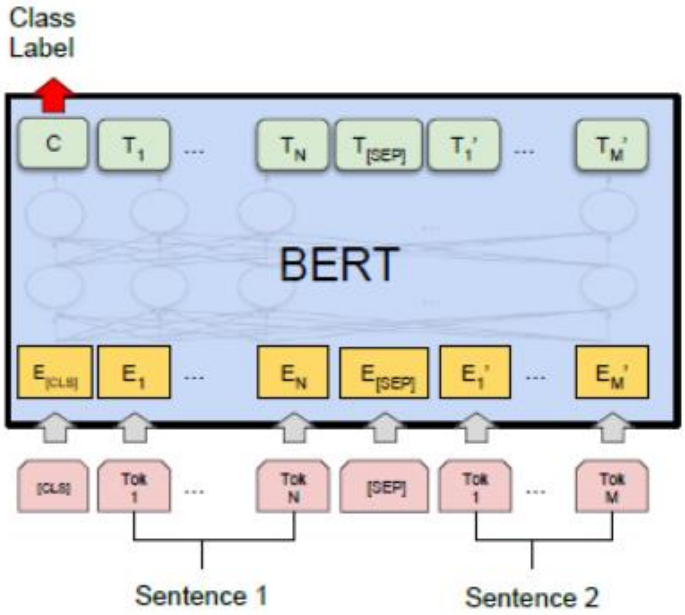


Figure 4.3: BERT model for Sentence Pair classification tasks (Devlin, Chang, Lee, & Toutanova, 2018).

4.4 Experiments

4.4.1 Dataset

The MADE challenge contains 27328 annotated relations (Yu, Jagannatha, Liu, & Liu, 2018), such as the relation between Indication and Drug entities, where the medication has been prescribed as a direct treatment for the Indication entity. There are seven types of relations between two different entities (see Fig. 4.4) as follows:

- Adverse: [Drug] caused [ADE]
- Reason: [Drug] given for [Indication/Reason]
- Dosage: [Drug] has [Dosage]
- Frequency: [Drug] has [Frequency]
- Duration: [Drug] has [Duration]
- Manner/Route: [Drug] has [Route]
- Severity: [Sign/Symptom and another Disease (SSD)] has [Severity]

We created different models with the training set of 833 clinical notes of MADE dataset (Jagannatha & Yu, 2016; Yu, Jagannatha, Liu, & Liu, 2018). We randomly split the training dataset into 15% and 85% for training and development set, respectively. The models were evaluated on the test dataset composed of 126 clinical notes, and we calculated the mean precision, recall, and F1 measure for all type of relations (see Fig. 4.4).

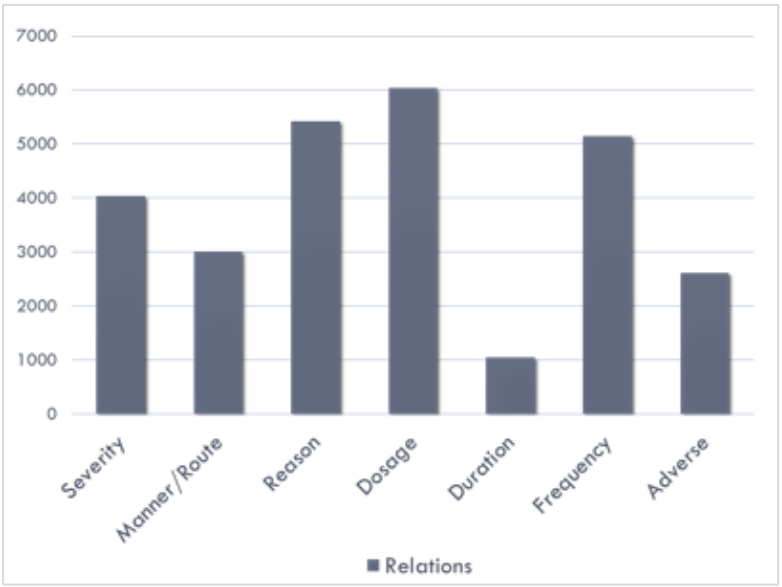


Figure 4.4: Number of annotations for every type of relation in MADE dataset (Jagannatha et al., 2018)

4.4.2 Experimental settings

We experimented with multiple hyper-parameter settings on the development set, different hidden layer sizes of LSTM (100, 200, 300) and learning rate (initial at 1e−2, 1e−3, 1e−4) adjusted by Adam (Adaptive Moment Estimation (Kingma & Ba, 2014)) algorithm for learning rate optimization. We use Dropout (at 0.5) to prevent over-fitting.

The word embedding size provided by FastText is 300 (pre-trained with unlabelled data from Wikipedia) (Bojanowski, Grave, Joulin, & Mikolov, 2017), and the more accurate sequence length was 30 words.

4.5 Results and Discussion

We report performance metrics Precision, Recall and F1 of our best run for each model (see Table 4.2). The results analysis is centred on F1 score because it combines precision and recall, mainly on the micro-averaged F1 score, an aggregate F1 score over all classes (Jagannatha, Liu, Liu, & Yu, 2019).

Relation	Recall	Precision	F1	Mean Distance (#char ±SD)
Severity	0,665	0,816	0,733	5 ±34
Manner	**0,960**	0,890	0,924	18 ±25
Reason	0,582	0,828	0,684	**96** ±164
Dosage	0,947	**0,933**	**0,940**	11 ±22
Duration	0,932	0,640	0,759	20 ±27
Frequency	0,830	0,827	0,828	25 ±30
Adverse	0,683	0,700	0,691	82 ±**187**
Overall	0,779	0,831	**0,804**	36,7

Table 4.2: Performance of LSTM+Features model with Test dataset of MADE Challenge

Our best model is LSTM with addition of external features, which increased in 12.3% F1 of the baseline model (LSTM alone) and the extraction of Adverse relations in 11.8%. The model also got better results than the BERT model (see Table 4.3). Contextual information provided by LSTM was not enough to determine the correct relations, due to the separation by several sentences between two candidate entities, so there are no words that inform explicitly the relation. In those cases of relations between entities separated by long distances, the provision of other features (such as the distances) becomes crucial to reinforce the model when LSTM does not receive the necessary connection of words to predict accurately the relation between the entities involved. The external features provided another relevant type of information that improved the accuracy of LSTM, indicating the effectiveness of combine deep-learning models and knowledge features.

Our best F1 score (0.804 in overall) is lower in 0.064 when it is compared with the model based on Random Forest (Chapman A. B., Peterson, Alba, DuVall, & Patterson, 2018), which was the best model reported in MADE Challenge (see Table 4.3), and 0.068 respect to LSTM-based model proposed in (Dandala, Joopudi, & Devarakonda, 2019). The model obtains high F1 score on categories such as Manner and Dosage of medications, but the model struggled on Reason and Adverse relation types. We obtained 0.691 F1 for Adverse relation, and the model with the best overall performance (Attentional LSTM) obtained just 0.660 in F1, but it was more accurate with Reason relations (see Table 4.4). Meanwhile, the second best model (Random Forest based) obtained 0.720 for the Adverse relation. The model based on Random Forest like (Chapman A. B., Peterson, Alba, DuVall, & Patterson, 2018) do not take in account the interaction between words (as LSTM), so they can include bi-grams or trigrams to reduce the lack of this information, which create a massive number of features and consequently the models can be over-fitted (Huynh, He, Willis, & Rüger, 2016). On the other hand, the LSTM model with knowledge systems (Dandala, Joopudi, & Devarakonda, 2019) uses heavily hand-engineered features usable only for a specific type of relations, which is not easily reproducible for Adverse relations.

Model	**Recall**	**Precision**	**F1-score**
LSTM (baseline)	0.668	0.772	0.716
LSTM+Features	0.779	0.831	0.804
BERT (fine-tuning)	0.484	0.134	0.210
Random Forest (Chapman A. B., Peterson, Alba, DuVall, & Patterson, 2018)	**0.881**	0.857	0.868
Attention LSTM (Dandala, Joopudi, & Devarakonda, 2019)	0.855	**0.888**	**0.872**

Table 4.3: Performance metrics for the relation extraction task (best two state-of-the-art models in *italics*)

The performance for Adverse relations is the lowest between all type of relations for all the models (see Table 4.4), it is due to the low number of samples for Adverse relations respect the other type of relations (see Fig. 4.4), and also due to the text span between two entities in this relation is longer, the mean distance between ADE-Drug entities is 82 characters with the highest Standard Deviation SD (see Table 4.2) (Jagannatha, Liu, Liu, & Yu, 2019), meanwhile the distance is much smaller in other relations like Duration-Drug, where it is just 20 characters (four times less than Adverse relation), and Duration-Drug relation get more F1 even with less than half of

annotations of Adverse relation (see Table 4.4). Another issue that affects the accuracy is related with the named entities involved in the ADE-Drug relation, the Drug names usually would not be confused with other categories, but ADE has a common vocabulary with Indication and SSD categories, like headache or fever, which can reduce the accuracy for relations with these entities.

Relation	**LSTM**	**LSTM +Feature**	**Random Forest** (Chapman A. B., Peterson, Alba, DuVall, & Patterson, 2018)	**Attention LSTM** (Dandala, Joopudi, & Devarakonda, 2019)	**% total Training ann.**
Severity	0,699	0,733	0.952	0.940	15
Manner	0,798	0,924	0.923	0.953	11
Reason	0,577	0,684	0.742	0.809	20
Dosage	0,818	0,940	0.961	0.942	22
Duration	0,734	0,759	0.834	0.878	4
Frequency	0,745	0,828	0.934	0.935	19
Adverse	0,618	0,691	**0.720**	0.660	9
Overall	0,716	0,804	0.868	**0.872**	23165 total ann.

Table 4.4: Performance (F1) with Test dataset of MADE Challenge

In a real scenario given some clinical note, the entity recognition and relation identification is carried out in row, thus we join both tasks in a pipeline to detect the entities and their relations in the raw data. This full system for NER and Relation Extraction has been evaluated on the MADE test dataset of Relation Extraction, called the joint NER-RI task (3).

The NER pre-trained model provides the input to the Relation Extraction model (see Fig. 4.5). Therefore, we get a propagation error because the NER system provides both True Positive and False Positive entities as input for the second model. Then the NER performance is the same as in Table 3.7 (see Subsection 3.4.1), and we see the expected performance reduction compared with Relation Extraction task module alone (see Table 4.4), due to propagation error mentioned before.

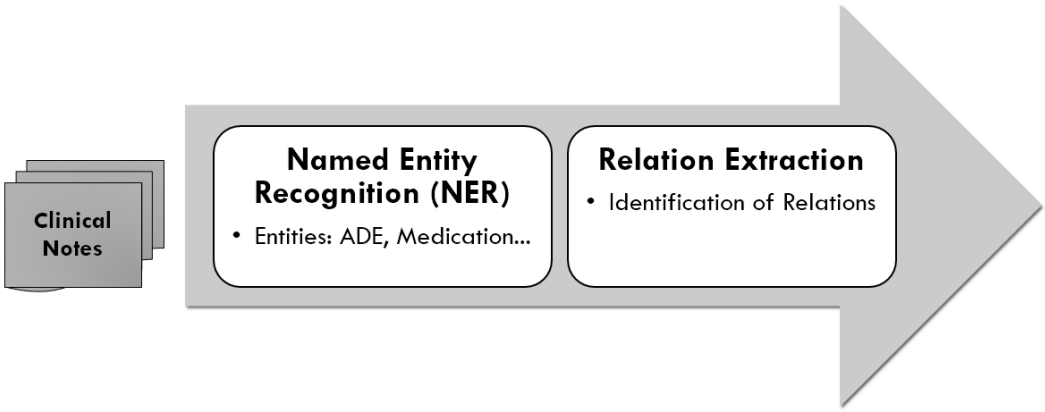


Figure 4.5: Pipeline for Joint task

Table 4.5 presents our results in this joint relation identification task of MADE challenge (shared task 3), where the Relation Extraction models are based on Random Forest, SVM or LSTM. We are 0.06 F1 points below the best model reported in (Jagannatha, Liu, Liu, & Yu, 2019), and two models using Random Forest for Relation Extraction have different results, because the NER model is only CRF in the model with lower performance (see Table 4.5). The Relation Extraction model fails immediately if just one of the two entities that conform a relation is False Negative of the NER model or classified in a wrong type of entity.

Model	Recall	Precision	F1-score
NER>>LSTM+Features	0.566	0.548	0.557
NER>>Random Forest*	0.435	0.643	0.519
NER>>SVM*	0.601	0.597	0.599
CRF>>Random Forest*	0.518	**0.692**	0.592
NER>>Attention LSTM*	**0.632**	0.603	**0.617**

Table 4.5: Comparison with MADE challenge task 3

Note: *Results collected by (Jagannatha, Liu, Liu, & Yu, 2019), NER are models based on LSTM+CRF

4.6 Conclusion

We investigates whether Deep Learning approaches can be effectively used for relation extraction of Adverse Drug Reactions in clinical notes. We could achieve comparable results with state-of-the-art models, and we show the importance of training Deep Learning (Bi-LSTM) model with additional external features. The features are relevant mostly for implicit relations where LSTM does not receive enough information to identify the relation between the entities involved. The external features provided

another essential type of information that improved the accuracy of LSTM, indicating the effectiveness of combined vectors of features and contextual knowledge of the relations.

We got similar performance to the best model in the joint task of relation identification, thus our full system based in Deep Learning is able to classify entities and its relations. The performance to extract Adverse Drug Reactions (Adverse relation) was closed to the best models, however, it is still low mainly due to the long distance between the entities that participate in the relation (ADE and Medication). Therefore, as future work is essential to extract other types of features, to recognize the implicit connection between entities separated by several sentences.

CHAPTER 5

Real Life Scenario

5.1 Introduction

The data mining allows to exploit huge amounts of clinical records collected by hospitals throughout patient's life, in order to discover information such as new Adverse Drug Events. In this chapter we describe a real scenario of Pharmacovigilance (Drug Safety) with data in French from consultations carried out by general practitioners (text with minor pre-processing) (Gazzotti, Faron-Zucker, Gandon, Lacroix-Hugues, & Darmon, 2019), where is necessary to process the raw data due to its natural issues, such as medical jargon and acronyms, unknown vocabulary particular to medical field, besides the often use of abbreviations by doctors.

Annotations to clinical notes can be obtained using Dictionary-based methods available for medical text in French. We can compare Dictionary-based methods only with our model for Named Entity Recognition task, because they are not able to extract relations between entities. These methods have been evaluated through a gold-standard medical corpus in French provided with annotations, in a health challenge where supervised methods based on Machine Learning have been also evaluated.

5.2 Raw Clinical Data

The data source provided by the Medicine Faculty of Université Côte d'Azur is called PRIMEGE (Regional Information Platform in General Medecine) (Lacroix-Hugues, Darmon, Pradier, & Staccini, 2017). PRIMEGE is a database that contains anonymous data in French from about 40000 patients collected directly from consultation software (Electronic Health Records), with no effort of doctors to feed the database for research purpose. It contains both structured text (with codes) and notes in free text (unstructured), currently data of 13 GPs (general practitioners) about patient’s health collected from 2012 to 2016 (see original description in Fig. 5.1).

A procedure was carried out on PRIMEGE for transforming free text in CISP2 codes (*Classification Internationale des Soins Primaires*, in French) (Lacroix-Hugues, Darmon, Pradier, & Staccini, 2017), the annotation allowed to associate most reasons of encounter and diagnostics with this International Classification of Primary Care (ICPC), that classify text in categories like Symptoms, Infections, Injuries and Congenital Anomalies. The validation of the annotation procedure have been performed by comparing the codes obtained with those found in ECOGEN (*Étude des Éléments de la Consultation en Médecine Générale*) and CISMef (University of Rouen) for the same labels (Lacroix-Hugues, Darmon, Pradier, & Staccini, 2017). Missing data is an important limit to exploit Electronic Health Records. However, some incomplete data could be reconstituted by automatic cross-referencing of prescriptions, laboratory results and CISP2 codes in PRIMEGE.

Catégorie de données		Données à recueillir
Caractéristiques du médecin		Sexe, année de naissance Ville et code postal
Caractéristiques du patient	Générales	Année de naissance, ville, code postal, sexe Affection longue durée (O/N), Médecin traitant CMU / AME
	Statut socio-professionnel	Profession, nombre d'enfants, situation familiale Catégorie socio-professionnelle
	ATCD et Facteurs de risque	Antécédents familiaux : libellé, lien familial, codes CISP2/CIM10 Antécédents personnels : libellé, date de début et de fin, codes CISP2/CIM10 Facteurs de risque : libellé, date de début et de fin, consommation par jour (tabac), codes CISP2/CIM10
Caractéristiques de la consultation	Générales	Date de la consultation, numéro de consultation / visite
	Episode de soins	Motifs de la consultation (libellé et code CISP2) Symptômes Données de biométrie (TA, poids …) Observation médicale Diagnostics (libellé et codes CISP2/CIM10) Médicaments (nom, posologie, motifs de prescription, nombre de boîtes, nombre de renouvellement, remarque, traitement de fond (O/N), générique (O/N), Code CIP) Prescriptions paramédicales (type (kiné, infirmier …) et libellé, motif de la prescription) Examens complémentaires (type (biologie/imagerie) et libellé, résultat, motif de la prescription) Procédures effectuées lors de la consultation (procédure, résultat)

Figure 5.1: Description of data collected in PRIMEGE (Lacroix-Hugues, Darmon, Pradier, & Staccini, 2017)

The entity–relationship model of PRIMEGE database is centred on Visit (consultations) to the doctor (see Fig. 5.2), Visit entity is linked to patient's data like Prescriptions, Drugs, Diagnoses and unstructured text called Observations (i.e. clinical notes). We focus our work on the Observations that contain richer data such as Adverse Drug Events, medical observation and symptoms, diagnoses, medications, reasons of encounter, radiology results, weight, blood pressure, etc. (see Fig. 5.3).

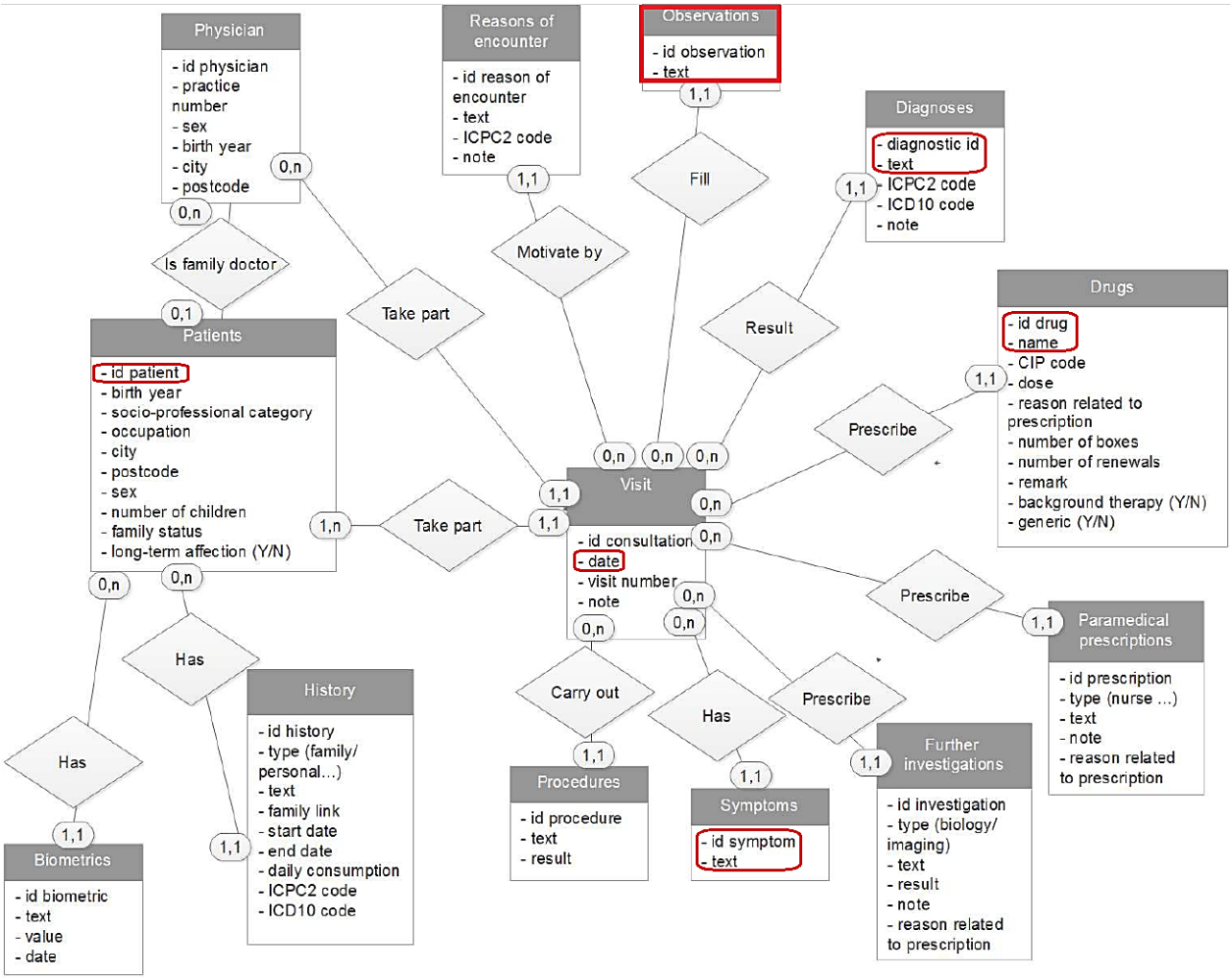


Figure 5.2: Entity–relationship model of PRIMEGE database (Lacroix-Hugues, Darmon, Pradier, & Staccini, 2017)

Volume de données

Elément	Nombre
Patients	38 970
Consultations	241 472
Antécédents	146 333
Données de biométrie	218 371
Motifs de consultation	157 009
Diagnostics	109 181
Lignes de prescriptions médicamenteuses	560 536
Symptômes recueillis	11 638
Procédures de soin	8 146
Examen complémentaires	546 262
Prescriptions paramédicales	11 048
Observations/notes	36 702

Figure 5.3: Number of elements in PRIMEGE (Lacroix-Hugues, Darmon, Pradier, & Staccini, 2017)

5.3 Clinical Notes Pre-processing

The PRIMEGE clinical notes only have passed through data anonymization procedure (non pre-processing), we found there different problems besides the particular vocabulary to the medical field. The main problems are the use of medical acronyms, e. g. TA, AB, TDR, ADP, ASD, MT, RC, EFR, many misspelling words, e. g. *apetit* instead *appétit*, *esport* instead *sport*. Moreover the clinical notes contain abbreviations and medical jargon, for example:

cardio instead *cardiologue, cardiologique, cardiomégalie* or *cardiopathie.*
gastro instead *gastroenterite, gastrolenterolgue, gastroscopie* or *gastrocnemiens.*
pulm instead *pulmonaire, cardiopulm* or *cardiopulmonaire.*
nl=normal, g.=gauche, qq=quelques, dte=droite, trt=traitement, tr=trouble.

When we used a standard tool to solve these problems, we added more errors due to the medical vocabulary that us unknown for these tools like NLTK python library. Then we corrected the clinical text adding a domain dictionary. PRIMEGE clinical notes have been corrected with the support of commercial software called Antidote 9, by a Master student in Computer Science (Delwende, 2018). The tool is able to detect misspelling, abbreviations and missing punctuation. The tool gives suggestions for each error, and the correction adopted was verified manually because the automatic selection sometimes is wrong. Misspelling errors were corrected sending batches of 1000 lines to

the tool, but the following abbreviations were replaced manually because the tool is not able to do it.

med : médecin médical paramédical médicamenteux
trt,ttt :traitement
dps: depuis
j : jours
RV=rendez-vous
RAS=rien a signaler
sem:semaine
gé=généraliste
tjr: toujours
chir:chirurgie
dmde=demande
pb: problème
dl,doleur:douleur
qq.=quelque

5.4 Automatic annotators based on ontologies

Dictionary-based approaches have been used in biomedical domain due to the formal vocabulary available in dictionaries, e. g. the automatic annotators from BioPortal, CISMEF and LIRMM, although these dictionary-based approaches are limited because they are only capable to detect concepts presented in ontologies (or terminologies). Ontology is a formal naming and definition of types, properties, and interrelations (clearly defined) of the entities (concepts) that exist for a particular domain, which is widely accepted by its community (Gruber, 1993). Annotators can exploit ontologies to match concepts with data provided by users.

The BioPortal developed by the National Center for Biomedical Ontology (NCBO), provides biomedical ontologies and tools to search and visualize them. BioPortal includes ontologies with UMLS codes (Unified Medical Language System) that specify semantic types such as Disease. An example of medical ontologies is CIM-10 (CIM version 10), the *Classification Internationale des Maladies* (International Statistical Classification of Diseases, ICD), this classification contains codes for diseases, signs and symptoms, abnormal findings, complaints, social circumstances, etc.

We received support from CISMef (Catalog and Index of French Language Medical Sites) that belong to the University of Rouen (Cabot, Soualmia, Dahamna, & Darmoni, 2016), in order to use the ECMT (Extracting Concepts with Multiple Terminologies)

tool (Pereira, et al., 2008), which annotates raw text using the concepts of health ontologies or terminologies (in French and English). They have SOAP and REST web services to provide a response in XML for each concept extracted of the text, it contains the health concept, the identifier and its semantic type if the health concept is included in the UMLS Meta-thesaurus. Figure 5.4 shows an example of ECMT web service for annotations, where *Terme* is the preferred Term (between several synonyms) for the entity found in the text, Ter is the Terminology acronym, Code is the internal code of the terminology, and CUI (Concept UMLS Identifier) is the Unified Medical Language System (UMLS) code.

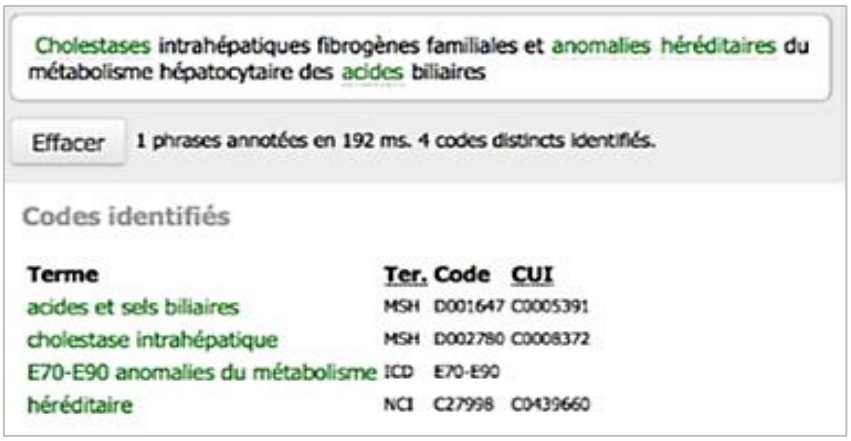


Figure 5.4: ECMT annotation through web service

ECMT access to terminologies available in HeTOP (Health Terminology-Ontology Portal) repository (Cabot, Soualmia, Dahamna, & Darmoni, 2016). HeTOP hosts more than 55 Terminology-Ontology in several languages, mostly the French version of ontologies in English collected by NCBO BioPortal, e.g. MeSH and ICD-10. HeTOP contains original dedicated ontologies for drug terminology called "*Racines des Médicaments*" (PHA) and adverse events mentions called WHO-ART (Adverse Reaction Terminology).

An example of a query looking for *Fievre* (Fever) within WHO-ART (Adverse Reaction Terminology) ontology:

WHO-ART (Main Class)

ETAT GENERAL (Class)

FIEVRE (Preferred Term). Terms included (Synonyms):

FIEVRE D'ORIGINE MEDICAMENTEUSE

PYREXIE

REACTION FEBRILE

A simple annotation needs at least three elements, the position of the entity (words) identified in the text, the words and the label annotated. ECMT gives other data in the annotations like ontology acronym inside the CISMeF internal code, and preferred Term for the entiy, as in the following samples:

[Begin, End, Entity, Label, #Annotation, **Ontology**, Preferred Term]

d+ nuque et au dessus de l oreille , mieux avec advil , surveillance 3J . oreille nl

[49,54,"advil",CHEM,1,**PHA**_RAC_178,ADVIL]

se plaint de fluctuation d'anxiété

[27,34,"anxiété",PHYS,1,**ART**_HT_0166_HLT,ANXIETE]

rhinorhée post , apyrexie pas de perte d'apetit – surveillance

{[0,9,"rhinorhée",DISO,1,**ART**_IT_0539_IT10,RHINORRHEE],

[33,47,"perte d'apetit",DISO,2,**ICD**_SC_R630,ANOREXIE]}

We obtained annotations of PRIMEGE notes using the ECMT tool (Pereira, et al., 2008). PRIMEGE contains 46422 notes with 1413030 tokens (words). We chose ontologies to identify adverse events mentions, medications and diseases; they are called WHO-ART (Adverse Reaction Terminology), PHA (*Racines des spécialités pharmaceutiques françaises*) and ICD-10 (International Statistical Classification of Diseases). Then ECMT identified seven type of entities in the PRIMEGE notes according to the UMLS categories, ACTIvities and behaviors, CHEMical and drugs, DISOrders, PHENomena, PHYSiology, PROCedures, CONCepts and ideas (see Table 5.1). For instance, the text "*sensation bizarre*" has been found in WHO-ART ontology with the Disorder label (CISMEF code ART_IT_0171_IT5). This annotations and notes could be the training dataset for supervised learning approaches.

Category	Annotations	Ontology
ACTivities	272	All
CHEMical	24225	PHA
DISOrder	52403	All
PHENomena	295	All
PHYSiology	796	All
PROCedures	17	ICD
CONCepts	2	ICD
Total	78010	

Table 5.1: UMLS type of entities identified by ECMT in PRIMEGE notes

We also used the annotator of LIRMM BioPortal (Jonquet, 2019), thanks to the collaboration a LIRMM researcher from University of Montpelier, we extracted concepts on PRIMEGE (46422 notes) using the LIRMM annotator on both original and corrected text. We added all the ontologies available like WHO-ART (Adverse Reaction Terminology), and ICD-10, with the focus on five semantic groups; ACTIvities, CHEMical and drugs, DISOrders, PHENomena, PHYSiology (see Table 5.2). We increased the total number of annotations obtained from the original clinical notes in approximately 29% with the corrected PRIMEGE clinical notes.

Category	Original	Corrected
ACTivities	6986	8445
CHEMical	22674	24654
DISOrder	146495	196215
PHENomena	4149	5276
PHYSiology	10848	12809
Total	191152	247399

Table 5.2: Number of annotations of original and corrected PRIMEGE clinical notes

Samples of original clinical notes of PRIMEGE annotated by ECTM and LIRMM (see Fig. 5.5), shows less annotations with LIRMM mostly for medications (CHEM). We verified for example the medication *advil* is not recognized due to the annotator only search for the formal name, advil400mg, it does not take into account synonyms.

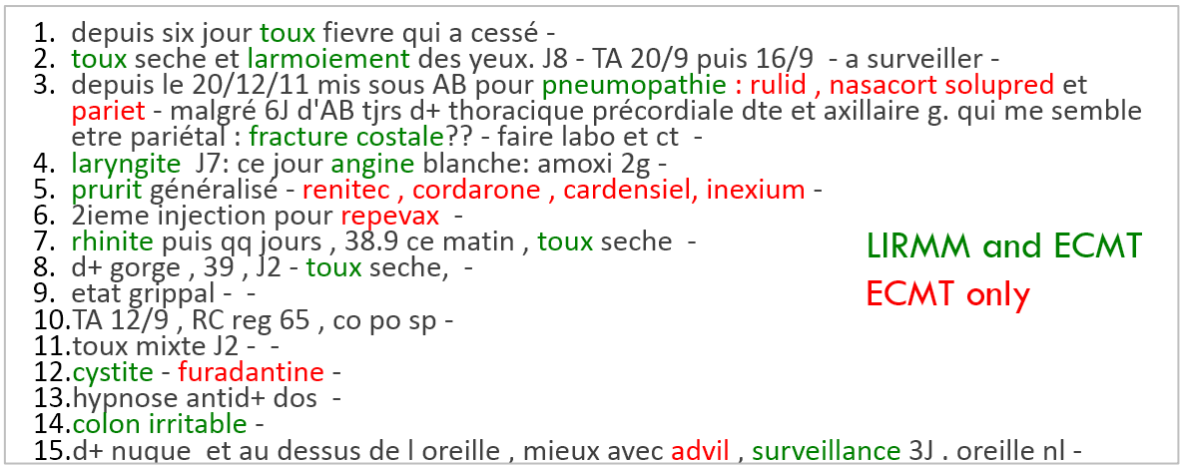


Figure 5.5: Samples of annotations of PRIMEGE

5.5 NER for Medical Text in French

We only could compare the Dictionary-based methods with our model for Named Entity Recognition task, because they are not capable to extract relations between entities. For this purpose, we use same data presented in CLEF (Conference and Labs of the Evaluation Forum) challenges, which proposes evaluation laboratories for information access systems every year. The CLEF eHealth Evaluation Lab contains Information Extraction tasks in Medical domain, they allow to compare different approaches like dictionary-based for NER in French medical text. They used a corpus in French for Medical Entity Recognition called QUAERO in the CLEF eHealth Lab 2015 (Task 1b) (Névéol, et al., 2015) and Lab 2016 (Task 2) (Névéol, et al., 2016). The last years of CLEF eHealth Lab, the NER task used other corpus to extract causes of death from death certificates.

QUAERO French Medical Corpus has manually annotations for ten categories of medical entities defined in UMLS (Névéol, Grouin, Leixa, Rosset, & Zweigenbaum, 2014); ANATomy, CHEMical and drugs, DEVices, DISOrders, GEOGraphic areas, LIVing Beings, OBJeCts, PHENomena, PHYSiology, PROCedures. They built a corpus with titles of MEDLINE papers, database of the US National Library of Medicine (NLM). MEDLINE dataset contains 833 files in both training and test dataset (see Table 5.3), splitting made in CLEF challenge 2015 (task 1b) (Névéol, et al., 2015).

Categories	Training set	Test set
Anatomy	495	510
Chemical	346	341
Devices	39	35
Disorders	963	982
Geographic	34	51
Living Beings	297	324
Objects	27	38
Phenomena	60	49
Physiology	190	159
Procedures	573	607
Total Ann.	2994	3094
Total tokens	10.500	10.800
N° files	833	833

Table 5.3: Number of annotations by category

The CLEF challenge task 1b (2015) received the submission of several teams. They have plain and normalized entity recognition subtasks for each corpus (MEDLINE and EMEA) (Névéol, et al., 2015).The knowledge-based method of ECMT was used by CISMeF team, with default settings of the web service and seven French medical terminologies and ontologies (Knowledge Organization Systems, KOS). ECMT seeks to match terms listed in Knowledge Organization Systems to the corpus. They participated again in CLEF 2016 (see Table 5.5) using the name SIBM team (Cabot, Soualmia, Dahamna, & Darmoni, 2016).

Other teams used SVM and CRF based methods, for example, Watchdogs team used CRF on stemmed tokens with standard lexical features and the word position in the sentence. LIMSI team presented a system based on the combination of three classifiers, in order to deal with embedded entities (16% of entities in training set), a CRF detects non-embedded entities, other context-free CRF detects embedded entities, finally, SVM classifier identifies their semantic class, with token ngrams, morphologic features, and dictionary consultation in language-dependent external sources (Névéol, et al., 2015). The best team (Erasmus) used a Dictionary-based concept recognition system, with automatic translation of English UMLS terms to index the QUAERO corpus. They kept the best performance in the CLEF eHealth 2016 (see Table 5.5) (Névéol, et al., 2016), using post-processing steps to reduce the number of false positives (FP). Most of the teams reported results only for the plain entity recognition subtask (Névéol, et al.,

2015), Table 5.4 shows runs submitted by the teams for NER (exact match) on the MEDLINE test corpus (CLEF 2015).

Method	Team	TP	FP	FN	Precision	Recall	F-measure
Dictionary-based concept recognition system, with automatic translation of english UMLS terms	Erasmus-run1	1861	756	1116	0.711	0.625	**0.665**
	Erasmus-run2	1912	886	1065	0.683	**0.642**	0.662
Binary classifiers with the same sets of features: uni-grams and bi-grams and associated information	*IHS-RD-run1-fix*	*1195*	*1782*	*376*	*0.761*	*0.401*	*0.526*
	IHS-RD-run2	1188	383	1789	**0.756**	0.399	0.522
CRF on stemmed tokens with standard lexical	Watchdogs-run1	1215	490	1762	0.713	0.408	0.519
Combination of classifiers CRF and SVM	LIMSI-run1	1121	834	1856	0.573	0.377	0.455
CRF models for each entity type and corpora	HIT-WI Lab-run1	1068	671	1909	0.614	0.359	0.453
CRF on stemmed tokens with standard lexical	Watchdogs-run2	1364	2069	1613	0.397	0.458	0.426
ECMT: Knowledge-based using 7 french ontologies	CISMeF-run1	680	4412	2297	0.134	0.228	0.169
SVM classifiers based on a distant learning approach	*UPF-run1-fix*	*189*	*2788*	*817*	*0,064*	*0,188*	*0,095*
Binary classifiers with the same sets of features	IHS-RD-run1	75	168	2902	0.309	0.025	0.047
SVM classifiers based on a distant learning approach	UPF-run1	82	888	2895	0.085	0.028	0.042
	average (official)				0.498	0.355	0.396
	average-fix				0.553	0.396	0.440
	median (official)				0.594	0.388	0.454
	median-fix				0.649	0.400	0.487

Table 5.4: Results for entity recognition task in CLEF eHealth 2015 (Névéol, et al., 2015)

Team	TP	FP	FN	Precision	Recall	F-measure
*Erasmus-run3.unofficial**	*2220*	*1045*	*881*	***0.680***	***0.716***	***0.698***
Erasmus-run1*	2139	1330	962	0.617	0.690	0.651
Erasmus-run2*	2103	1273	998	0.623	0.678	0.649
SIBM-run2*	1357	761	1745	0.641	0.438	0.520
SIBM-run1*	1476	1258	1626	0.540	0.476	0.506
BITEM-run1*	1376	1032	1741	0.571	0.442	0.498
LITL-run1*	998	556	2105	0.642	0.322	0.429
LITL-run2	989	561	2114	0.638	0.319	0.425
*UPF-run2.unofficial**	*969*	*5050*	*2138*	*0.161*	*0.312*	*0.212*
UPF-run1*	736	5053	2369	0.127	0.237	0.166
UPF-run2	739	5050	2367	0.128	0.238	0.166
average				0.503	0.426	0.446
median				0.617	0.438	0.498

Table 5.5: Results for entity recognition task in CLEF eHealth 2016 (Névéol, et al., 2016)

We used our final model (described in Subsection 3.3.2) to get results with MEDLINE dataset (see Fig. 5.6). We split each document into separate sentences with variable length. The training set is divided in 90% for training and 10% for validation set. The embedding layer of the model is a pre-trained W2V in French of 200 dimensions created by FastText (Bojanowski, Grave, Joulin, & Mikolov, 2017). We found the optimal set up of hyper-parameters by adjusting the hyper-parameters values during different runs to get the maximum accuracy, which is 70 sequence length, 200

cells in the bidirectional LSTM layer and initial learning rate 0.001 using Adam optimizer. We implemented the BIO tagging, then we got 21 labels instead the eleven original labels (ten categories plus None) in the inference layer (CRF), which are reshape to the original labels during the post-processing to save the annotations in BRAT format.

Model(input=sequence_words, output=sequence_labels)

Layer (type)	Output Shape	# Parameters
embedding_1 (EmbeddingFR)	(None, 70, 200)	1162600
dropout_1 (Dropout)	(None, 70, 200)	0
bidirectional_1 (Bi-LSTM)	(None, 70, 200)	240800
crf_1 (CRF)	(None, 70, 21)	4704

Total parameters: 1,408,104
Trainable parameters: 245,504
Non-trainable parameters: 1,162,600

Figure 5.6: The best model for NER in MEDLINE dataset

Our model overcomes the best result presented in CLEF 2015 with 0.671 F1 on test set (see Table 5.6). The teams provided runs with slightly different results in CLEF 2016 than previous year (more average and mean performance for all teams), our model is only overcame by an unofficial result presented by Erasmus team that used a Dictionary-based concept recognition system, and their official results obtained lesser F1 than us. The second best result was the dictionary based model of ECMT, with 0.520 F1 obtained by their team called SIBM (CISMEF team in 2015). Approaches based in Machine Learning models (CRF and SVM) obtained worst results such as UPF and LIMSI teams. We observe less performance (see Table 5.6) in categories with low number of samples in training set (devices, geographic areas, objects, phenomena and physiology).

	TP	FP	FN	Precision	Recall	F1
Anatomy	306	132	146	0.6986	0.6770	0.6876
Chemical	215	91	96	0.7026	0.6913	0.6969
Devices	7	4	26	0.6364	0.2121	0.3182
Disorders	580	129	184	0.8181	0.7592	0.7875
Geographic	17	16	33	0.5152	0.3400	0.4096
Living Beings	200	84	104	0.7042	0.6579	0.6803
Objects	6	11	32	0.3529	0.1579	0.2182
Phenomena	6	13	42	0.3158	0.1250	0.1791
Physiology	55	64	97	0.4622	0.3618	0.4059
Procedures	325	149	233	0.6857	0.5824	0.6298
Overall	1717	693	993	0.7124	0.6336	**0.6707**

Table 5.6: Performance for plain entity recognition on MEDLINE test set

The current model was trained with text in English, therefore it must be trained again with annotated text in French to analyse PRIMEGE notes (see Fig. 5.7), but we only have a training set of PRIMEGE with annotations for medical entities such as medications and disorders (without ADE), and the dataset do not have ADR relations because dictionary-based annotators cannot extract it.

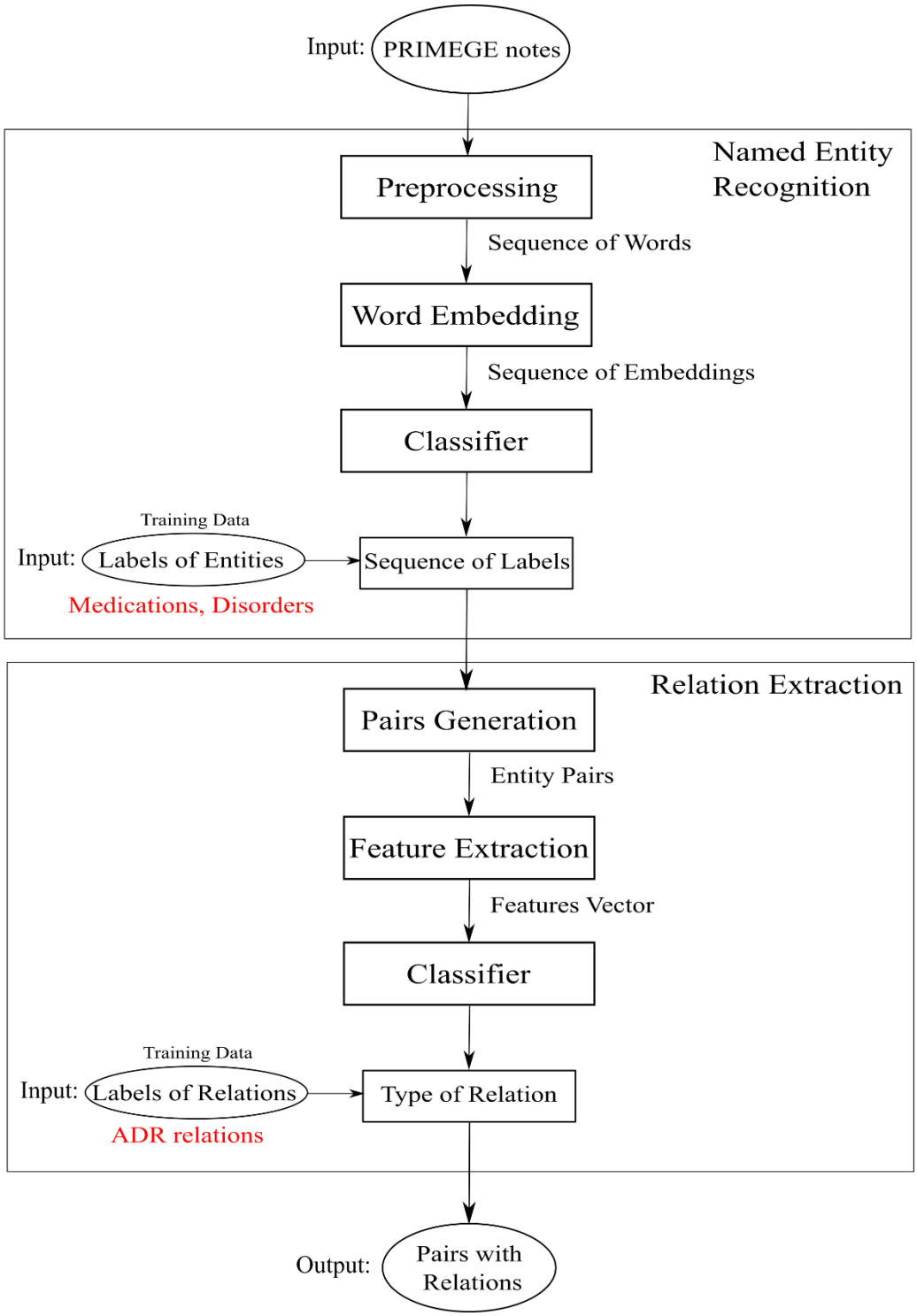


Figure 5.7: Full model for ADR detection on PRIMEGE dataset

5.6 Conclusion

We exploits clinical notes provided in Electronic Health Records (PRIMEGE database in French) to evaluate the real Pharmacovigilance scenario. There is quite necessary to pre-process clinical notes due to misspelling, medical jargon, acronyms and abbreviations, what we did with a tool that was manually supported. We create annotations of medical entities to PRIMEGE clinical notes using Dictionary-based methods available for medical data in French, the LIRMM and ECMT annotators based on ontologies.

We could compare our model against these type of annotators for the Named Entity Recognition task of medical data in French (CLEF eHealth Lab). Our model based on Deep Learning got higher accuracy than the official results of CLEF eHealth challenges, which evaluated dictionary based approaches (such as ECMT) and Machine Learning such as CRF and SVM.

In the real life scenario, we must work on supervised models capable of misspelling correction, in addition, learn to replace acronyms and medical abbreviations found in the raw clinical notes. Then, we need the support of automatic annotators and domain experts to obtain the desired annotations for our supervised models, in order to detect relations between specific entities.

CHAPTER 6

Conclusions and Future Work

6.1 Conclusions

Clinical notes contain rich information such as medical observations, diagnoses, medications and the information required for our surveillance of adverse side effects. We have presented models for automatic detection of Adverse Drug Reactions in clinical notes, which rely on the supervised approach for identification of entities and relations between the entities. We divided the problem into Named Entity Recognition and Relation Extraction tasks, then we trained the models with labelled data of clinical notes. The models exploit contextual information in the sentences and features of entities and relations in order to enrich their representations. The global contribution of this thesis is the model for identifying medical relations (such as Adverse Drug Reactions), given clinical notes as input, the model returns pairs of entities that have a relation.

The information extraction has been performed by supervised learning methods that overcome the limitations of other methods such as dictionary based models. We provide a Named Entity Recognition model to recognize medical entities on clinical notes. The input of word sequences must provide relevant information for the supervised model, thus we used representations for words such word embeddings. The results are better with FastText embedding than others models trained with word2vec (Skip-gram algorithm). We also built character-level features extracted with another LSTM, which was used in conjunction with word embeddings as a comprehensive word representation. This conjunction of features increased the performance of LSTM, and we reached the best performance achieved for the NER task adding a CRF layer, because CRF considers the dependency between chains of successive labels that is not taken into account by LSTM.

Moreover, we work with clinical notes in other language (Spanish) different to English, we tried different word representations to increase the performance of our best

model. The best model learned the embedding during training, probably due to the Out-of-Vocabulary (OOV) problem on pre-trained embeddings, which do not have word representation for entities such as proteins found in the test set.

The trained NER model can identify and annotate medical entities on other clinical notes. Then we developed other supervised approach for Relation Extraction from the recognized entities. The results show the importance of additional external features for models based on LSTM neural networks. The features are relevant mostly for implicit relations or long distance relations where LSTM does not receive any contextual information to identify the relation between the entities involved. The features provided another essential type of information that improved the accuracy of the baseline (LSTM alone), which indicates the effective combination of feature vectors and contextual knowledge of the relations.

The joint model or full system is able to identify entities and its relations. The performance to extract Adverse Drug Reactions (Adverse relations) was similar to the best models of the state of the art, but it is low mainly due to the long distance between the entities that participate in that type of relation (ADE and Medication entities). Therefore, it is important to extract other type of features in order to recognize implicit connections between entities separated by many words. We could use the supervised models on different labelled datasets with minor adaptations, then this approach can be applied to other domains with annotated data, which facilitates the future work with the models (available online as open source).

In the real life scenario of Pharmacovigilance, we got raw clinical notes of the PRIMEGE database in French, which contain relevant data such as ADE. The pre-processing of clinical notes is quite necessary due to misspelling, medical jargon, acronyms and abbreviations. We create annotations of medical entities to PRIMEGE clinical notes using Dictionary-based methods available for medical data in French, the LIRMM and ECMT annotators based on ontologies. We could compare our model against these type of annotators for Named Entity Recognition of medical data in French, the model obtained higher accuracy than the official results of the CLEF eHealth challenges, which evaluated dictionary based approaches (ECMT) and Machine Learning such as CRF and SVM. Then, domain experts could obtain the required

annotations for the supervised models, in order to detect relations between specific entities.

6.2 Future Work

In order to increase the accuracy, the model can be extended through additional layers of Transfer Learning models, using pre-trained language models as a feature extraction layer such as Multi-Task Deep Neural Network (MT-DNN) and XLNet models (see Fig. 6.1). They are large models of millions of parameters trained on big datasets (that are growing more every year), which improve state-of-the-art on many Natural Language Processing task.

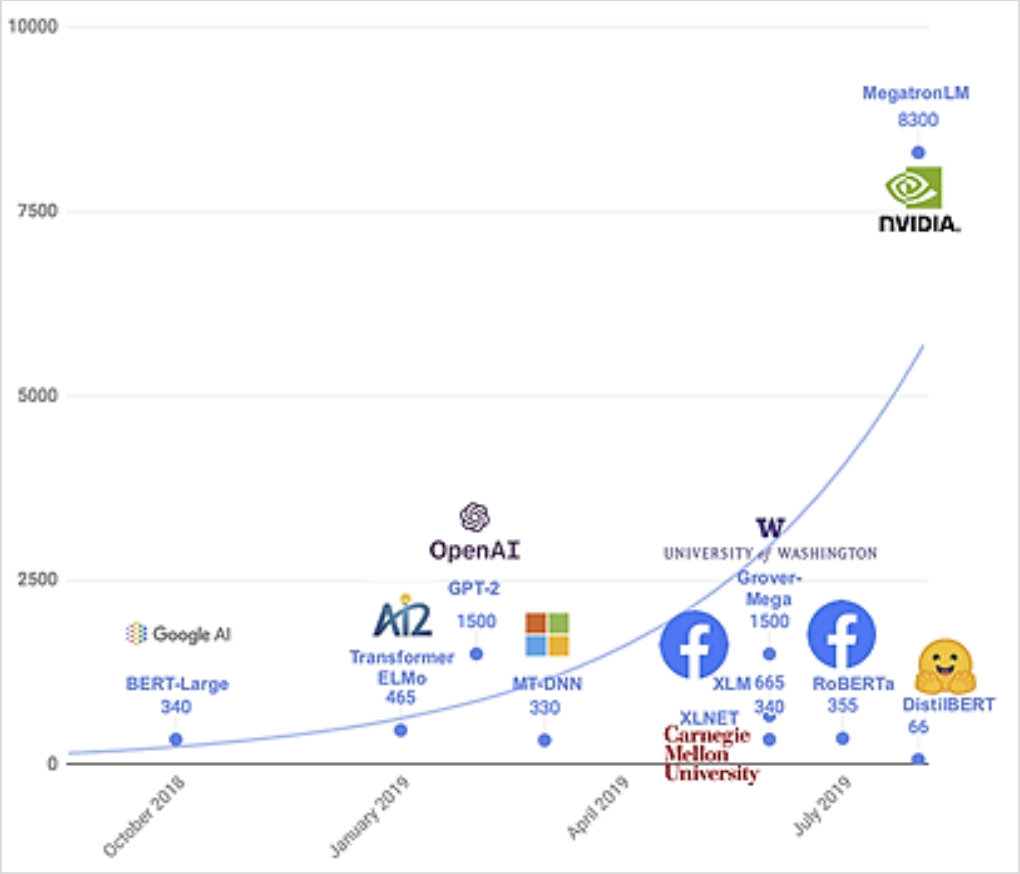


Figure 6.1: Language models with size in millions of parameters[1]

In order to reduce the Out-of-Vocabulary (OOV) problem of pre-trained embeddings, which do not have word representation for entities of special vocabulary such as medications, we suggest to create a dedicated word embedding for medical text, with

[1] https://medium.com/huggingface/distilbert-8cf3380435b5

the specific vocabulary for medications, protein, chemical names, etc. Moreover, to avoid the propagation error of the data pipeline, it is necessary to develop joint approaches to extract entities and relations between them simultaneously.

Most of the related research has been carried out on labelled data in English, so it would be important to develop language independent methods, for languages without annotations for training such as ADE entities in French. On the contrary, it is necessary to get (manual) annotations of clinical data in French, in order to tag new medical reports and identifying ADR relations in them. The model has to train in the same way we did it with annotated clinical notes in English and Spanish. The current purpose is to feed a database of ADR, sorted by number of occurrences, to compare with the known side effects of the medications related to every ADR.

This information could be provide to pharmacovigilance centers such as the *Centre Regional De Pharmacovigilance* of Nice[2], which work for identification, evaluation and prevention of side effects risk or other medication-related problems. This information also could feed Decision Support System for treatment prescriptions during medical consultations, to alert about potential side effect according to the patient's clinical history.

[2] https://extranet.chu-nice.fr/centre-pharmacovigilance

Appendices

Appendix A. Description of the best model for Relation Extraction

Model(input=[inputwords, featuresVector], output=relations)

Layer (type)	Output Shape	Param #	Connected to
inputwords_1 (InputLayer)	(None, 30)	0	
embedding_1 (Embedding)	(None, 30, 300)	3934500	inputwords_1[0][0]
bidirectional_1 (Bidirectional)	(None, 200)	320800	embedding_1[0][0]
featuresVector (InputLayer)	(None, 200)	0	
concatenate_1 (Concatenate)	(None, 400)	0	bidirectional_1[0][0] featuresVector[0][0]
dense_1 (Dense)	(None, 8)	3208	concatenate_1[0][0]

Total params: 4,258,508
Trainable params: 324,008
Non-trainable params: 3,934,500

Appendix B. Example of Features Vector

- Sentence: *The patient has <Severity>significant</Severity> <ADE>peripheral neuropathy</ADE> secondary to <Drug>velcade</Drug>.*
- Entity types and text of Candidate Entities for each relation:
  Relations: X_1. Severity_type X_2. Adverse (ADR)
  Entities: SEVERITY : DRUG ADE : DRUG
  Text: *significant:Velcade* *peripheral neuropathy:Velcade*
- Type and number of entities between candidates
  X_1: 1 entity, Types: ADE X_2: 0 entity, Types: --
- Number of words and sentences between candidates
  X_1: 1 sentence, 4 words X_2: 1 sentence, 2 words
- Feature_dictionary= $[X_1, X_2, \dots, X_t]$ =

[{num_sentences_between: 1, num_entities_between: 1, text_in_anno1: 'velcade', second_entity_type:<**SEVERITY**>, text_in_anno2: 'significant', first_entity_type:<**DRUG**>, entities_between:<ADE>, num_tokens_between: 4},

{num_sentences_between: 1, num_entities_between: 0, text_in_anno1: 'peripheral neuropathy', second_entity_type:<**DRUG**>, text_in_anno2: 'Velcade', first_entity_type:<**ADE**>, entities_between:<>, num_tokens_between: 2}]

- Features Vector**:** Transform feature_dictionary to array =
  [[1, 0, 0, ... , 1, 1, 4],
  [0, 1, 0, … , 1, 0, 2]]

Bibliography

Agirre, A. G., Marimon, M., Intxaurrondo, A., Rabal, O., Villegas, M., & Krallinger, M. (2019). Pharmaconer: Pharmacological substances, compounds and proteins named entity recognition track. *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*, (pp. 1-10).

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research, 3*, 1137-1155.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks, 5*, 157-166.

Bodenreider, O. (2004). The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research, 32*, 267-270.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics, 5*, 135-146.

Budi, I., & Bressan, S. (2003). Association rules mining for name entity recognition. *Proceedings of the Fourth International Conference on Web Information Systems Engineering, 2003. WISE 2003.*, (pp. 325-328).

Cabot, C., Soualmia, L. F., Dahamna, B., & Darmoni, S. J. (2016). SIBM at CLEF eHealth Evaluation Lab 2016: Extracting Concepts in French Medical Texts with ECMT and CIMIND. *CLEF (Working Notes)*, (pp. 47-60).

Chapman, A. B., Peterson, K. S., Alba, P. R., DuVall, S. L., & Patterson, O. V. (2018). Hybrid system for adverse drug event detection. *International Workshop on Medication and Adverse Drug Event Detection*, (pp. 16-24).

Chapman, A. B., Peterson, K. S., Alba, P. R., DuVall, S. L., & Patterson, O. V. (2019). Detecting adverse drug events with rapidly trained classification models. *Drug safety, 42*, 147-156.

Chiu, J. P., & Nichols, E. (2016). Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics, 4*, 357-370.

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. *Proceedings of the 25th international conference on Machine learning*, (pp. 160-167).

Dandala, B., Joopudi, V., & Devarakonda, M. (2018). IBM Research System at MADE 2018: detecting adverse drug events from electronic health records. *International Workshop on Medication and Adverse Drug Event Detection*, (pp. 39-47).

Dandala, B., Joopudi, V., & Devarakonda, M. (2019). Adverse drug events detection in clinical notes by jointly modeling entities and relations using neural networks. *Drug safety*, *42*, 135-146.

Delwende, B. (2018). Deep spell check. Technical Report at I3S research lab.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Florez, E., Precioso, F., Pighetti, R., & Riveill, M. (2019). Deep Learning for Identification of Adverse Drug Reaction Relations. *Proceedings of the 2019 International Symposium on Signal Processing Systems*, (pp. 149-153).

Florez, E., Precioso, F., Riveill, M., & Pighetti, R. (2018). Named entity recognition using neural networks for clinical notes. *Proceedings of the International Workshop on Medication and Adverse Drug Event Detection*, (pp. 7-15).

Gazzotti, R., Faron-Zucker, C., Gandon, F., Lacroix-Hugues, V., & Darmon, D. (2019). Injecting Domain Knowledge in Electronic Medical Records to Improve Hospitalization Prediction. *European Semantic Web Conference*, (pp. 116-130).

Genthial, G. (2017). Sequence Tagging with Tensorflow. *GitHub repository*. Retrieved from https://guillaumegenthial.github.io/sequence-tagging-with-tensorflow.html

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM networks. *Proceedings of the IEEE International Joint Conference on Neural Networks*, *4*, pp. 2047-2052.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, *5*(2), 199-221.

Gurulingappa, H., Mateen-Rajpu, A., & Toldo, L. (2012). Extraction of potential adverse drug events from medical case reports. *Journal of biomedical semantics, 3*, 15.

Gurulingappa, H., Rajput, A. M., Roberts, A., Fluck, J., Hofmann-Apitius, M., & Toldo, L. (2012). Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of biomedical informatics, 45*, 885-892.

Hauben, M., & Bate, A. (2009). Decision support methods for the detection of adverse events in post-marketing data. *Drug discovery today, 14*, 343-357.

Hochreiter, S., & Schmidhuber, J. (1997). LSTM can solve hard long time lag problems. *Advances in neural information processing systems*, (pp. 473-479).

Honnibal, M. (2015). NLTK Library. *https://www.nltk.org/api/nltk.tag.html*, (p. last accessed 2018/03/10).

Huynh, T., He, Y., Willis, A., & Rüger, S. (2016). Adverse drug reaction classification with deep neural networks., (pp. 877-887).

Jagannatha, A. N., & Yu, H. (2016). Bidirectional RNN for medical event detection in electronic health records. *Proceedings of the conference. Association for Computational Linguistics. North American Chapter. Meeting, 2016*, p. 473.

Jagannatha, A. N., & Yu, H. (2016). Structured prediction models for RNN based sequence labeling in clinical text. *Proceedings of the conference on empirical methods in natural language processing. conference on empirical methods in natural language processing, 2016*, p. 856.

Jagannatha, A., Liu, F., Liu, W., & Yu, H. (2019). Overview of the first natural language processing challenge for extracting medication, indication, and adverse drug events from electronic health record notes (MADE 1.0). *Drug safety, 42*, 99-111.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *European conference on machine learning*, (pp. 137-142).

Jonquet, C. (2019). *Ontology Repository and Ontology-Based Services--Challenges, contributions and applications to biomedicine & agronomy*. Ph.D. dissertation.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lacroix-Hugues, V., Darmon, D., Pradier, C., & Staccini, P. (2017). Creation of the First French Database in Primary Care Using the ICPC2: Feasibility Study. *Studies in health technology and informatics, 245*, 462-466.

Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Li, L., Jin, L., Jiang, Z., Song, D., & Huang, D. (2015). Biomedical named entity recognition based on extended recurrent neural networks. *2015 IEEE International Conference on bioinformatics and biomedicine (BIBM)*, (pp. 649-652).

Li, Q., Deleger, L., Lingren, T., Zhai, H., Kaiser, M., Stoutenborough, L., . . . Solti, I. (2013). Mining FDA drug labels for medical conditions. *BMC medical informatics and decision making, 13*, 53.

Liao, W., & Veeramachaneni, S. (2009). A simple semi-supervised algorithm for named entity recognition. *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, (pp. 58-65).

Liu, Z., Yang, M., Wang, X., Chen, Q., Tang, B., Wang, Z., & Xu, H. (2017). Entity recognition from clinical texts via recurrent neural network. *BMC medical informatics and decision making, 17*, 67.

Liwicki, M., Graves, A., Fernàndez, S., Bunke, H., & Schmidhuber, J. (2007). A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. *Proceedings of the 9th International Conference on Document Analysis and Recognition, ICDAR 2007*.

Luo, L., Yang, Z., Yang, P., Zhang, Y., Wang, L., Lin, H., & Wang, J. (2018). An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition. *Bioinformatics, 34*, 1381-1388.

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Magge, A., Scotch, M., & Gonzalez-Hernandez, G. (2018). Clinical NER and relation extraction using bi-char-LSTMs and random forest classifiers. *International Workshop on Medication and Adverse Drug Event Detection*, (pp. 25-30).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2017). Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*.

Mikolov, T., Yih, W.-t., & Zweig, G. (2013). Linguistic regularities in continuous space word representations. *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, (pp. 746-751).

Mintz, M., Bills, S., Snow, R., & Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, (pp. 1003-1011).

Munkhdalai, T., Liu, F., & Yu, H. (2018). Clinical relation extraction toward drug safety surveillance using electronic health record narratives: classical learning versus deep learning. *JMIR public health and surveillance, 4*, e29.

Nadeau, D. (2007). *Semi-supervised named entity recognition: learning to recognize 100 entity types with little supervision.* Ph.D. dissertation, University of Ottawa.

Nadeau, D., Turney, P. D., & Matwin, S. (2006). Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. *Conference of the Canadian society for computational studies of intelligence*, (pp. 266-277).

Névéol, A., Cohen, K. B., Grouin, C., Hamon, T., Lavergne, T., Kelly, L., . . . others. (2016). Clinical information extraction at the CLEF eHealth evaluation lab 2016. *CEUR workshop proceedings, 1609*, pp. 28-42.

Névéol, A., Grouin, C., Leixa, J., Rosset, S., & Zweigenbaum, P. (2014). The QUAERO French medical corpus: A ressource for medical entity recognition and normalization. *In proc biotextm, reykjavik.*

Névéol, A., Grouin, C., Tannier, X., Hamon, T., Kelly, L., Goeuriot, L., & Zweigenbaum, P. (2015). CLEF eHealth Evaluation Lab 2015 Task 1b: Clinical Named Entity Recognition. *CLEF (Working Notes).*

Nikfarjam, A., & Gonzalez, G. (2011). Pattern mining for extraction of mentions of adverse drug reactions from user comments. *AMIA annual symposium proceedings, 2011*, pp. 1019-1026.

Nikfarjam, A., Sarker, A., O'connor, K., Ginn, R., & Gonzalez, G. (2015). Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *Journal of the American Medical Informatics Association, 22*, 671-681.

Pawar, S., Palshikar, G. K., & Bhattacharyya, P. (2017). Relation extraction: A survey. *arXiv preprint arXiv:1712.05191*.

Peng, N., Poon, H., Quirk, C., Toutanova, K., & Yih, W.-t. (2017). Cross-sentence n-ary relation extraction with graph lstms. *Transactions of the Association for Computational Linguistics, 5*, 101-115.

Pereira, S., Névéol, A., Kerdelhué, G., Serrot, E., Joubert, M., & Darmoni, S. J. (2008). Using multi-terminology indexing for the assignment of MeSH descriptors to health resources in a French online catalogue. *AMIA Annual Symposium Proceedings, 2008*, p. 586.

Poibeau, T., & Kosseim, L. (2001). Proper name extraction from non-journalistic texts. In *Computational Linguistics in the Netherlands 2000* (pp. 144-157). Brill Rodopi.

Quirk, C., & Poon, H. (2016). Distant supervision for relation extraction beyond the sentence boundary. *arXiv preprint arXiv:1609.04873*, 1171-1182.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE, 77*, 257-286.

Ramshaw, L. A., & Marcus, M. P. (1999). Text chunking using transformation-based learning. *Natural language processing using very large corpora*, 157--176.

Rokach, L., & Maimon, O. Z. (2008). *Data mining with decision trees: theory and applications* (Vol. 69). World scientific.

Roth, D., & Yih, W.-t. (2004). *A linear programming formulation for global inference in natural language tasks.* Tech. rep., ILLINOIS UNIV AT URBANA-CHAMPAIGN DEPT OF COMPUTER SCIENCE.

Sarker, A., Ginn, R., Nikfarjam, A., O'Connor, K., Smith, K., Jayaraman, S., . . . Gonzalez, G. (2015). Utilizing social media data for pharmacovigilance: a review. *Journal of biomedical informatics, 54*, 202-212.

Settles, B. (2004). Biomedical named entity recognition using conditional random fields and rich feature sets. *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, (pp. 107-110).

Sienčnik, S. K. (2015). Adapting word2vec to named entity recognition. *Proceedings of the 20th nordic conference of computational linguistics, nodalida 2015, may 11-13, 2015, vilnius, lithuania*, (pp. 239-243).

Swampillai, K., & Stevenson, M. (2011). Extracting relations within and across sentences. *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, (pp. 25-32).

Tang, B., Cao, H., Wu, Y., Jiang, M., & Xu, H. (2013). Recognizing clinical entities in hospital discharge summaries using Structural Support Vector Machines with word representation features. *BMC medical informatics and decision making, 13*, p. S1.

Uzuner, Ö., Solti, I., & Cadag, E. (2010). Extracting medication information from clinical text. *Journal of the American Medical Informatics Association, 17*, 514-518.

Wang, X., Hripcsak, G., Markatou, M., & Friedman, C. (2009). Active computerized pharmacovigilance using natural language processing, statistics, and electronic health records: a feasibility study. *Journal of the American Medical Informatics Association, 16*, 328-337.

Xu, D., Yadav, V., & Bethard, S. (2018). UArizona at the MADE1. 0 NLP Challenge. *Proceedings of machine learning research, 90*, 57-65.

Yu, H., Jagannatha, A., Liu, F., & Liu, W. (2018). NLP Challenges for Detecting Medication and Adverse Drug Events from Electronic Health Records.

Zhang, S., & Elhadad, N. (2013). Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. *Journal of biomedical informatics, 46*, 1088-1098.

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., & Xu, B. (2016). Attention-based bidirectional long short-term memory networks for relation classification. *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, (pp. 207-212).