UNIVERSITY OF THE BASQUE COUNTRY (UPV/EHU)
Computer Systems and Languages Department

PhD dissertation

Adverse Drug Reaction extraction on Electronic Health Records written in Spanish

Sara Santiso González

2019

UNIVERSITY OF THE BASQUE COUNTRY (UPV/EHU)
Computer Systems and Languages Department

Adverse Drug Reaction extraction on Electronic Health Records written in Spanish

This is the report of the thesis "Adverse Drug Reaction extraction on Electronic Health Records written in Spanish", written by Sara Santiso under the supervision of Dr. Arantza Casillas and Dr. Alicia Pérez.

Bilbao (2019).

A mi padre y a mi madre

Acknowledgments

Primero me gustaría dar las gracias a mis directoras Alicia y Arantza por ayudarme siempre en todo lo que han podido, no solo durante la tesis, sino desde que empecé con la investigación en el Trabajo Fin de Grado y también durante el Trabajo Fin de Master. Gracias por todo el apoyo y el tiempo que me habéis dedicado.

También me gustaría dar las gracias a todos los compañeros del grupo IXA. A pesar de estar realizando la tesis en Bilbao, siempre he podido contar con su ayuda en todo lo que he necesitado. En especial me gustaría agradecer a Gorka, Igone, Koldo, Maite y Nora los comentarios que me hicieron durante la revisión de esta memoria, que fueron muy útiles para mejorarla.

Por último, quiero agradecer a mis padres todo el apoyo que me han dado durante este tiempo.

Official acknowledgments

This thesis was partially funded by the pre-doctoral grant (PRE_2015_1_0211, PRE_2016_2_0128, PRE_2017_2_0241, PRE_2018_2_0265) awarded by the Basque Government and also by the DETEAMI (Basque Goverment, 2014111003) and PROSAMED (Spanish Ministry of Science and Innovation, TIN2016-77820-C3-1-R) projects.

Abstract

This work focuses on the automatic extraction of Adverse Drug Reactions (ADRs) in Electronic Health Records (EHRs) written in Spanish. That is, our aim is to extract a response to a medicine which is noxious and unintended and occurs at doses normally used. From Natural Language Processing (NLP) perspective, this is approached as a relation extraction task in which the drug is the causative agent of a disease, the adverse reaction.

ADR extraction from EHRs involves major challenges. First, ADRs are rare events. That is, drugs and diseases found in an EHR are often unrelated or sometimes related as treatment, but seldom as ADRs. This implies the inference of a predictive model from samples with skewed class distribution. Second, EHRs are written by experts under time pressure, employing rich medical jargon together with colloquial expressions, not always grammatical, and it is not infrequent to find misspellings and both standard and non-standard abbreviations. All this leads to a high lexical variability.

To cope with these challenges, we explored several ADR detection algorithms and representations to characterize the ADR candidates. In addition, we assessed the tolerance of the ADR detection model to external noise such as the incorrect detection of the medical entities involved in the ADR extraction (drugs and diseases).

Contents

List of Figures

List of Tables

List of Abbreviations

AB-LSTM Attentive Bidirectional Long Short-Term Memory.

ADR Adverse Drug Reaction.

ATC Anatomical Therapeutic Chemical Classification System.

AUC Area Under ROC Curve.

Bi-LSTM Bidirectional Long Short-Term Memory.

Bi-RNN Bidirectional Recurrent Neural Network.

CBOW Continuous Bag-of-Words.

CNN Convolutional Neural Network.

CRF Conditional Random Fields.

CUI Concept Unique Identifier.

DARPA Defense Advanced Research Projects Agency.

DDI Drug-Drug Interaction.

DT Decision Trees.

EC European Commission.

EHR Electronic Health Record.

FFNN Feed Forward Neural Network.

FN False Negative.

FP False Positive.

FPR False Positive Rate.

GRU Gated Recurrent Units.

IAA Inter Annotator Agreement.

ICD International Statistical Classification of Diseases and Related Health Problems.

IF Impact Factor.

JCR Journal Citation Report.

LSTM Long Short-Term Memory.

ME Maximum Entropy.

MER Medical Entity Recognition.

NB Naive Bayes.

NER Named Entity Recognition.

NLP Natural Language Processing.

OOV Out-Of-Vocabulary.

PCA Principal Component Analysis.

POS Part-Of-Speech.

RF Random Forest.

RNN Recurrent Neural Network.

ROC Receiver Operating Characteristic.

SBWCE Spanish Billion Word Corpus and Embeddings.

SciELO Scientific Electronic Library Online.

SMOTE Synthetic Minority Oversampling TEchnique.

SNOMED CT Systematized Nomenclature of Medicine Clinical Terms.

SRS Spontaneous Reporting System.

SVM Support Vector Machines.

TN True Negative.

TP True Positive.

TPR True Positive Rate.

UMLS Unified Medical Language System.

WHO World Health Organization.

1 Introduction

1.1 Motivation

An Adverse Drug Reaction (ADR) is defined by the World Health Organization (WHO) as 'a response to a medicine which is noxious and unintended, and which occurs at doses normally used in man' (World Health Organization, 2002b). From this definition we can draw that the ADRs are difficult to avoid given that they happen when the medicine is taken correctly. Indeed, ADRs are the 4th to 6th largest cause for mortality in the USA (World Health Organization, 2002b). They result in the death of several thousands of patients each year, and many more suffer from ADRs. The percentage of hospital admissions due to ADRs was 11.5% in Norway, 13.0% in France and 16.0% in UK (World Health Organization, 2002b). This also entails costs of 15-20% of the hospital budget (World Health Organization, 2002b). In Spain, according to the "National study on hospitalisation-related adverse events (ENEAS)" developed by the Ministry of Health and Consumer Affairs (Ministerio de Sanidad y Consumo, 2006), an ADR is defined as 'alterations and/or injuries caused when the drugs are used appropriately, which are hardly preventable'. The ENEAS study indicates that 37.4% of the adverse events detected during the hospitalization were related to the medication.

The WHO informed about the importance of reporting ADRs to understand and treat the diseases caused by drugs and, as a result, improve the patients care (World Health Organization, 2002a). In fact, it was created VigiBase, the WHO global database of Individual Case Safety Re-

ports (Lindquist, 2008). The aim was to store the spontaneous reports in a international database to enable the earliest possible detection of drug-related problems. Other Spontaneous Reporting Systems (SRSs) were also created in different countries. For example, in USA the FDA Adverse Event Reporting System, in UK the Yellow Card System of Medicines and Healthcare products Regulatory Agency and in Spain the Spanish System of Pharmacovigilance of medicines for Human Use, which employs the database FEDRA developed by the Spanish Agency of Medicines and Medical devices. However, ADRs are still heavily under-reported, which makes their prevention difficult. Some of the reasons for under-reporting of ADRs are lack of time, different care priorities, uncertainty about the drug causing the ADR, difficulty in accessing reporting forms, lack of awareness of the requirements for reporting and lack of understanding of the purpose of SRSs (Hazell and Shakir, 2006). Note that, in these spontaneous reports, the ADRs have to be indicated manually by the medical experts. This process is not carried out automatically.

The creation of a system to automatically extract ADRs on Electronic Health Records (EHRs) would increase the reporting of the ADRs. Given that information stored digitally by the hospitals is growing, Natural Language Processing (NLP) techniques can be used to create a system that helps the doctors to analyze the ADRs of the patients in a given EHR, facilitating the decision making process and alleviating the work-load. As a consequence, the patients' health could improve and the pharmaco-surveillance service would be informed about the detected ADRs.

Precisely, research in the biomedical domain has attracted considerable attention in the last years in the NLP research community. Examples of this are workshops such as *BioNLP* (Demner-Fushman et al., 2018), *BioTxtM* (Limsopatham and Collier, 2016) and *Louhi* (Lavelli et al., 2018). *BioNLP* is interested in NLP for the biological and medical domains. *BioTxtM* focuses on NLP and text mining for biomedical and clinical text. *Louhi* is concerned about the automated processing of health documents. Moreover, we find the "Plan for the Advancement of Language Technology (Plan TL)" (Ministerio de Energía, Turismo y Agenda Digital, 2015) created by the Spanish government with the aim of developing the NLP and machine translation industries for Spanish and the co-official languages of Spain. The Plan TL involves different domains such as healthcare, education and tourism, reflecting that NLP can improve the quality and capacity of different public services. In the second Hackathon of the Language Technology organized in this plan, a new category for the biomedicine was introduced.

Due to the relevance that NLP has gained in the biomedical domain, several projects emerged at international level. In Europe, we find the "EU-ADR" project (Coloma et al., 2011), under the support of the European Commission (EC). The aim of this project is to exploit information from various EHR databases in Europe to produce a computerized integrated system for the early detection of ADRs. In USA, we find the "Big mechanism" program (Cohen, 2015), financed by the Defense Advanced Research Projects Agency (DARPA). The aim of this program is to develop technology to read research abstracts and papers to extract pieces of causal mechanisms, assemble these pieces into more complete causal models, and reason over these models to produce explanations. This program focuses mainly on the domain of cancer biology.

1.2 Framework

Our research was developed within the IXA group[1] of the University of the Basque Country (UPV/EHU), a multidisciplinary group composed of computer scientists and linguists among others. The IXA group combines linguistic modeling and data analysis with innovative probabilistic and machine learning approaches to NLP. To this group, several projects related with the medical domain and close to our work were granted by the Spanish Ministry of Science and Innovation (EXTRECM: TIN2013-46616-C2-1-R, PROSAMED: 2014111003) and the Basque Government (DETEAMI: 2014111003). These projects focus on the analysis of health records written in Spanish. This work was developed mainly within the framework of DETEAMI and PROSAMED projects:

- **DETEAMI**: *"Detección automática de efectos adversos a medicamentos en informes médicos usando tecnologías de procesamiento del lenguaje natural"* (meaning 'Automatic detection of adverse drug reactions in medical records using natural language processing techniques'). This project arises from the collaboration with different hospitals of the Basque Country. The aim is the development of a prototype to validate information extraction techniques for the medical domain. The hospital would be the client of this prototype. Our work is related with the fifth phase of the project. This phase consists in the creation of

[1] IXA group web page: http://ixa.si.ehu.eus/

a prototype for the detection of ADRs in EHRs generated for the patients. To this end, rule-based methods and machine learning methods would be employed.

- **PROSAMED**: *"PROcesamiento Semántico textual Avanzado para la detección de diagnósticos, procedimientos, otros conceptos y sus relaciones en informes MEDicos"* (meaning 'Advance semantic textual processing for the detection of diagnostic codes, procedures, concepts and their relationships in health records'). This project arises from the collaboration with research groups of other universities and institutions of the health system. The aim is the creation of solutions for processing medical texts. Our work is related with the fourth phase of the project. This phase consists in the implementation of a system to detect drug-disease pairs that correspond to ADRs of the patients in EHRs. To this end, machine learning (supervised as well as unsupervised methods) would be employed.

Within the framework of these projects and according to the consensus of medical experts, the ADR extraction was defined as a **relation extraction** task. That is, the aim is to detect ADR relations between the entities (drugs and diseases) recognized in a given text. For NLP, relation extraction is a crucial step towards natural language understanding applications (Bach and Badaskar, 2007). A fact that demonstrates the interest on relation extraction is the competitions created to this end. Some examples are *SemEval-2010 Task 8* (Hendrickx et al., 2010) and *SemEval-2018 Task 7* (Gábor et al., 2018). *SemEval-2010 Task 8* was created to obtain semantic relations between pairs of words, there were 9 semantic relations. *SemEval-2018 Task 7* was devoted to semantic relation extraction and classification in scientific papers, the relations could be classified in 6 categories.

Relation extraction consists in finding and classifying semantic relations among the text entities (Jurafsky and Martin, 2018). These relations can be binary, when two entities are involved, or complex, when multiple entities are involved (Zhou et al., 2014). In the medical domain there are defined some binary relations such as 'pharmacologic substance [causes] pathological function' or 'pharmacologic substance [treats] pathologic function' (Jurafsky and Martin, 2018). In this case, the binary relation that we wanted to find is similar to the one given in the first example. Specifically, we found the relation 'disease [caused-by] drug', that indicates a relation of type caused-by between the drug and the disease (e.g. '*astenia* [caused-by] *interferon*').

The NLP techniques that can be used for relation extraction tasks are divided in i) rule-based methods and ii) machine learning methods (Zhou et al., 2014). The machine learning methods, in turn, are divided in supervised and unsupervised methods (Dalianis, 2018). In this case we opted for supervised machine learning techniques, using EHRs with ADR annotations. The algorithm for relation extraction using supervised learning can be defined as in Figure 1.1 (Jurafsky and Martin, 2018). According to this, for relation extraction (FIND RELATIONS), first it is necessary to recognize the entities (FIND ENTITIES). In our case e_1 is the drug and e_2 is the disease. Finally, the relations created with these entities are classified (CLASSIFY RELATION) by assigning the label of the predicted class. In our case the relation is caused-by.

function FINDRELATIONS(*words*) **returns** *relations*

relations ← *nil*
entities ← FINDENTITIES(*words*)
forall **entity pairs** ⟨*e1*, *e2*⟩ **in** *entities* **do**
  if RELATED?(*e1*, *e2*)
    relations ← *relations*+CLASSIFYRELATION(*e1*, *e2*)

Figure 1.1: Relation extraction algorithm.

For the ADR extraction developed in this work, we distinguished the two steps involved in this task:

1. **Medical Entity Recognition (MER)** to find "drug" entities and "disease" entities. The "drug" entity encompasses either a brand name, a substance or an active ingredient and the "disease" entity encompasses either a disease, a sign or a symptom.

2. **ADR detection** to discover the relations between "drug" entities and "disease" entities that correspond to ADRs. The "drug" entity would be the causative agent and the "disease" entity would be the caused adverse reaction.

These steps can be developed using a **pipeline** approach as shown in Figure 1.2 for an extract from an EHR. First, the medical entities are recognized (e.g. the diseases '*fiebre de predominio vespertino*', '*sudoración*', '*astenia*' and the drug '*Interferon*') and then, the ADR relations are detected between

the previous entities (e.g. the pairs '*fiebre de predominio vespertino* - *Interferon*', '*sudoración* - *Interferon*', '*astenia* - *Interferon*'). This implies that ADR detection relies on the quality of the MER step. The ADR detection step explores the relations between drug-disease pairs, thus, an unrecognized entity might yield to undisclosed relations. Our work focused on the second step, the ADR detection, although we also explored the influence of the automatic recognition of medical entities in the detection of ADR relations.

In the ADR extraction process, we had to overcome some **challenges** that make this supervised classification task difficult. We can observe these in the EHR shown in Figure 1.3, from which we extracted the example shown previously. On the one hand, the ADRs are minority relations because generally the drug and the disease are either unrelated or related as treatment and, thus, the ADRs are rare cases. For example, the drug '*Interferon*' (meaning 'Interferon') was prescribed as a treatment for a disease different to the caused adverse reactions, '*VHC*' (meaning 'HCV'). On the other hand, the EHRs show multiple lexical variations. For example, the doctor refers to the drug '*Interferon*' (meaning 'Interferon') also with their abbreviation '*iFN*' or with the name '*iFN pegilado*', '*persistencia de la fiebre*' (meaning 'persistence of the fever') refers to the disease '*fiebre de predominio vespertino*' mentioned before and '*rash cutaneo*' (meaning 'skin rash') refers to the disease '*rash pruriginoso de predominio troncular consistente en máculas eritematosas*'. In addition, our EHRs are written in Spanish whereas the majority of biomedical NLP research has been done in English. The Spanish and other languages different to English count with few resources and tools to apply NLP in the medical domain. In this line, it is remarkable the recent interest in developing NLP tools for languages other than English (Névéol et al., 2018).

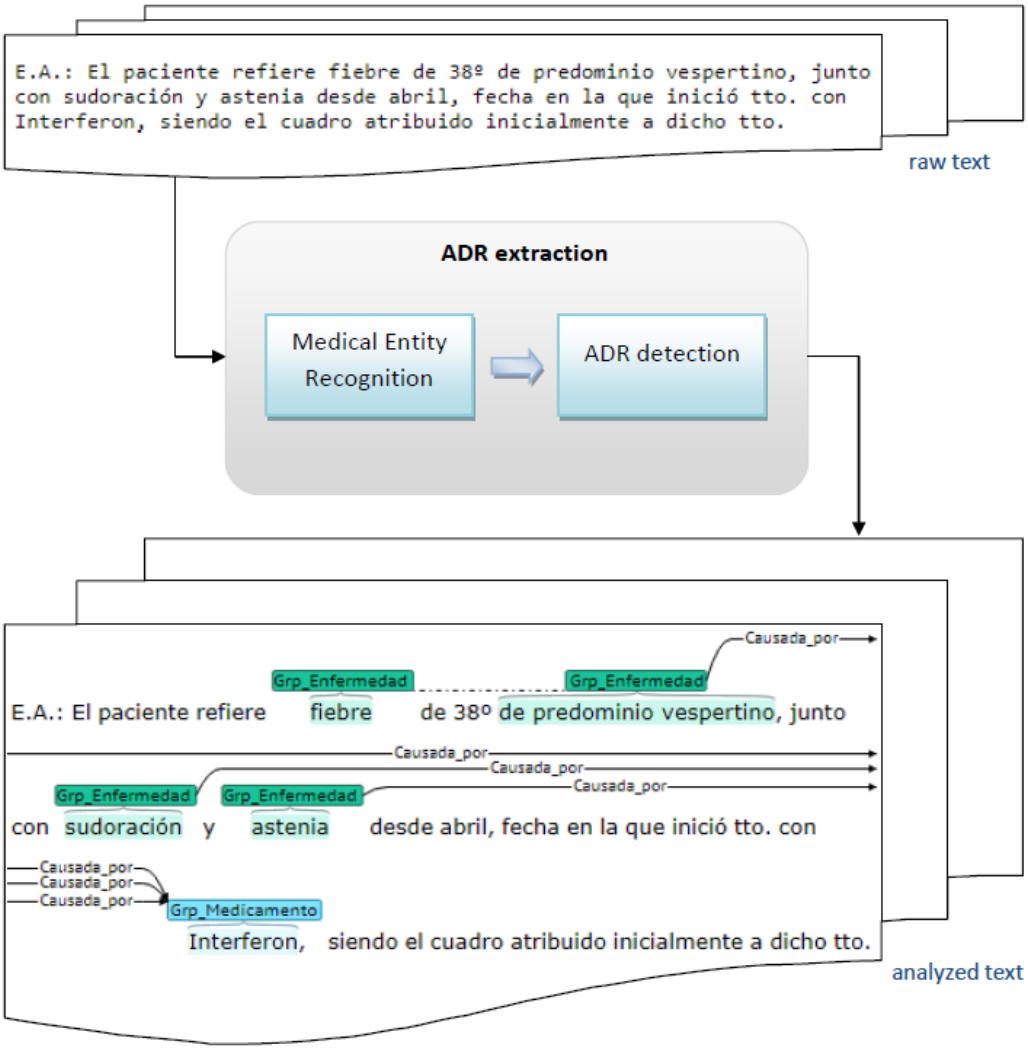


Figure 1.2: Scheme of the ADR extraction pipeline. The pipeline is applied to the sentence 'Current State: The patient has reported predominantly evening fever of 38°, together with sweating and weakness since April, date on which treatment with Interferon was started, being the symptoms attributed initially to this treatment.'.

Figure 1.3: Example of an annotated EHR.

1.3 Objectives and research questions

The main **objective** of this work is *the creation of a model able to detect automatically ADRs in EHRs written in Spanish.* This, in turn, encompasses the **sub-objectives** stated below:

- *Detect ADRs by discovering relations between the causative drug and the caused diseases.*
  The aim is to detect drug-disease pairs related as ADRs and not only the disease caused by the drug. The projects in which this work is framed (DETEAMI and PROSAMED) stated that indicating explicitly the entities involved in an ADR can result more useful for their study.

- *Discover approaches to overcome the class imbalance.*
  Given that ADRs are rare events, it is frequent to find the class imbalance problem in this task. Machine learning algorithms tend to expect balanced class distributions and learning the minority class is difficult for them. For this reason, our intention is to explore different techniques that could help to tackle this issue improving the ADR detection or find approaches that could be robust against imbalanced distributions of the class.

- *Discover robust representations to cope with the lexical variability and the data sparsity.*
  This is a challenge goal due to two factors. First, the EHRs are written during consultation time and each doctor uses different terms or expressions, producing lexical variations. Second, due to confidentiality issues, there is a lack of available EHRs. Then, our intention is to explore different representations in order to make the most of the annotated corpus.

According to the objective, the focus of this work can be summarized in the following **main research question**:

How can NLP techniques be applied to aid in the extraction of ADRs in EHRs written in Spanish?

Derived from this main research question, there are some **research questions** that we intend to answer. Unlike in other dissertations, we present these research questions along the chapters where each one emerges naturally, instead of in the introduction, in an attempt to facilitate their understanding.

1.4 Document structure

This dissertation is arranged in eight chapters. A brief description about the points addressed in each chapter is given below.

- Chapter 1 - Introduction

  In this chapter we introduce this work by explaining the motivation to develop the ADR extraction and the framework. We also present the objectives to achieve together with the main research question to address.

- Chapter 2 - Related work

  We make a review of the different works related with the ADR extraction task. We focus on the definition of ADR extraction, the techniques and features employed for the ADR classification, the corpora and the evaluation schemes used for ADR extraction.

- Chapter 3 - Experimental framework

  In order to reach a better understanding of the experiments, we present the corpora employed in our work. Furthermore, we describe the schemes and metrics employed for the evaluation of our systems.

- Chapter 4 - Adverse Drug Reaction detection with symbolic representations and Random Forest

  We describe the features employed to create the symbolic characterizations of the ADR events, our first approach. We present the Random Forest classifier used for ADR detection of intra-sentence as well as inter-sentence ADRs. We also explain the approaches explored to tackle the class imbalance.

- Chapter 5 - Adverse Drug Reaction detection with dense representations and Random Forest

  We explain the dense characterizations created from embeddings that were used together with the Random Forest classifier overcoming the class imbalance, our second approach. Moreover, we propose different smoothing techniques that were applied to the dense representations in order to improve the proximity between semantically related words.

- Chapter 6 - Adverse Drug Reaction detection with dense representations and Joint Attentive Bidirectional Long Short-Term Memory

  We change the classifier and we explain the neural networks used for ADR detection, including the core-features employed to infer the dense representations, as our third approach. We also present the techniques explored to overcome the class imbalance suited for neural networks.

- Chapter 7 - Tolerance of ADR detection to noise

  Until now, we just focus on the ADR detection step and we try different representations and classifiers. In this chapter we discuss the results obtained with the best performing approach, using slightly different corpora and incorporating the automatic detection of medical entities (the entire pipeline of Figure 1.2).

- Chapter 8 - Conclusions and future work

  Finally, we give the final conclusions, which include the response to the research questions and the main contributions. We explain the future lines of work regarding the ADR extraction. We also show the publications related to this work.

Apart from the chapters described above, we include appendices to explain some tasks developed in parallel and inherent to event extraction.

- Appendix A - Negated medical entity recognition

  We explain the two approaches explored to detect negated entities automatically. These negated entities are used to discard negative ADR candidates.

- Appendix B - Medical entity recognition

  We briefly explain some experiments developed to detect medical entities automatically. These entities are those used to observe the influence of MER step on ADR detection (see Figure 1.2).

- Appendix C - Detailed results: ADR detection with dense representations and Random Forest

  We give detailed results of the experiments developed in Chapter 5 for ADR detection using dense representations and the Random Forest classifier.

The abbreviations used throughout this document are expanded in page xxiii.

2 Related work

2.1 Introduction

In this chapter we review the different ways in which ADR extraction was tackled in related works. At the same time, we make a comparison and position our work with respect to previous approaches in terms of relevance. Our aim is to explain the differentiating factors and, above all, to align each work with ours. To begin with, we summarized in Table 2.1, related works in terms of the factors we found most differentiating to approach ADR extraction. Each factor, defined in depth in the forthcoming sections, is listed here:

i Definition of ADR extraction: Presence of the ADR in a piece of text (denoted as "P"), Mention of the ADR as an entity ("M"), Relation between the entities involved in the ADR ("R").

ii ADR classification techniques: Traditional ("T"), Deep Learning ("DL").

iii ADR characterization features: Symbolic ("S"), Dense ("D").

iv Corpora for ADR extraction: EHR ("E"), Social Media ("SM"), Scientific Publications ("SP"), Others ("O") as textual genres and English ("EN"), Japanese ("J"), Swedish ("SW") as languages.

v Evaluation of ADR extraction: Hold-Out ("HO") and k-fold Cross-Validation ("CV") as evaluation schemes. The F-measure ("F") of the

positive class obtained with hold-out is given as evaluation metric whenever possible; in other cases, the macro ("M") or micro ("μ") values or even the Area Under the ROC curve ("AUC") are given. In the table we always report the results corresponding to the best performing experiment.

Note that, for example, the definition of ADR itself is a conspicuous differentiating factor but also the characterization of ADRs, the approaches employed to extract them or the assessment techniques. Moreover, Table 2.1 just intends to summarize outstanding works but does not cope with all the works mentioned throughout this chapter. As an example, the table does not encompass those works that did not resort to supervised machine learning. By contrast, we found of interest to include not only works devoted to ADR extraction (in the top of the table) but also works within the medical domain in closely related tasks (in the middle) and relevant relation extraction approaches applied out of the medical domain (in the bottom).

Authors	Definition	Classification	Characterization	Corpora		Evaluation		
				Textual genre	Language	Scheme	Metric	Result
ADR EXTRACTION								
Aramaki et al. (2010)	R	T	S	E	J	10CV	F	59.8
Miura et al. (2010)	R	T	S	E	J	5CV	F	37.5
Sohn et al. (2011)	P	T	S	E	EN	HO	F	74.5
Botsis et al. (2011)	P	T	S	E	EN	HO	F_M	81.3
Gurulingappa et al. (2011)	P	T	S	SP	EN	10CV	F	76.0
Gurulingappa et al. (2012a)	R	T	S	SP	EN	HO	F	87.0
Karlsson et al. (2013)	P	T	S	E	SW	10CV	AUC	87.0
Patki et al. (2014)	P	T	S	SM	EN	10CV	F	65.2
Ginn et al. (2014)	P	T	S	SM	EN	10CV	F	76.6
Zhao et al. (2014)	P	T	S	E	SW	10CV	AUC	71.7
Zhao et al. (2015)	P	T	S	E	SW	10CV	AUC	76.3
Friedrich and Dalianis (2015)	P	T	S	E	SW	10CV	F	67.0
Li et al. (2015)	R	T	S	SP	EN	10CV	F	51.1

(Continued on next page)

Authors	Definition	Classification	Characterization	Corpora		Evaluation		
				Textual genre	Language	Scheme	Metric	Result
Sarker and Gonzalez (2015)	P	T	S	SP	EN	HO	F	81.2
Nikfarjam et al. (2015)	M	T	S,D	SM	EN	HO	F	82.1
Lin et al. (2015)	M	T	S,D	SM	EN	HO	F	62.5
Henriksson et al. (2015a)	R	T	S,D	E	SW	HO	F	27.2
Henriksson et al. (2015b)	P	T	D	E	SW	10CV	AUC	94.0
Zhang et al. (2016)	P	T	S,D	SM	EN	HO	F	54.9
Huynh et al. (2016)	P	DL	D	SP	EN	10CV	F	87.0
Stanovsky et al. (2017)	M	DL	D	SM	EN	HO	F	93.4
Lee et al. (2017)	P	DL	D	SM	EN	HO	F	64.5
Tutubalina and Nikolenko (2017)	M	DL	D	SM	EN	HO	F_M	79.8
Akhtyamova et al. (2017)	P	DL	D	SM	EN	HO	F	54.2
Cocos et al. (2017)	M	DL	D	SM	EN	HO	F	75.5
Gupta et al. (2018)	M	DL	D	SM	EN	HO	F	75.1
Wunnava et al. (2018)	M	DL	D	E	EN	HO	F	63.5
Masino et al. (2018)	P	DL	D	SM	EN	HO	F	45.7
Fabregat et al. (2018)	R	DL	D	SP	EN	10CV	F_M	75.6
MEDICAL DOMAIN								
Jagannatha and Yu (2016a)	M	DL	D	E	EN	10CV	F_μ	80.3
Jagannatha and Yu (2016b)	M	DL	D	E	EN	10CV	F_μ	86.3
Luo (2017)	R	DL	D	E	EN	HO	F_μ	77.5
Li et al. (2017)	R	DL	D	SP	EN	HO	F	66.1
Raj et al. (2017)	R	DL	D	E	EN	HO	F	64.4
Legrand et al. (2018)	R	DL	D	SP	EN	HO	F_M	83.9
He et al. (2019)	R	DL	D	E	EN	HO	F_μ	69.7
RELATION EXTRACTION APPLIED TO OTHER DOMAINS								
Celli (2010)	R	T	S	O	EN	10CV	F	26.7
Zeng et al. (2014)	R	DL	D	O	EN	HO	F_M	82.7
Ebrahimi and Dou (2015)	R	DL	D	O	EN	HO	F	82.7
Nguyen and Grishman (2015)	R	DL	D	O	EN	HO	F_M	82.8
Miwa and Bansal (2016)	R	DL	D	O	EN	HO	F_M	85.5
Zheng et al. (2016)	R	DL	D	O	EN	HO	F_M	83.8

(Continued on next page)

Authors	Definition	Classification	Characterization	Corpora		Evaluation		
				Textual genre	Language	Scheme	Metric	Result
Zhou et al. (2016)	R	DL	D	O	EN	HO	F_M	84.0
Katiyar and Cardie (2017)	R	DL	D	O	EN	HO	F_μ	55.9
Christopoulou et al. (2018)	R	DL	D	O	EN	HO	F_μ	64.2
Ren et al. (2018)	R	DL	D	O	EN	HO	F_M	87.4
Le et al. (2018)	R	DL	D	O	EN	HO	F_M	86.3

Table 2.1: Overview of the related works in chronological order, separating those devoted to ADR extraction (in the top), those within the medical domain (in the middle) and those out of the medical domain (in the bottom). The different values are: Presence (denoted as "P"), Mention ("M"), Relation ("R"), Traditional ("T"), Deep Learning ("DL"), Symbolic ("S"), Dense ("D"), EHR ("E"), Social Media ("SM"), Scientific Publications ("SP"), Others ("O"), English ("EN"), Japanese ("J"), Swedish ("SW"), Hold-Out ("HD"), Cross-Validation ("CV"), F-measure ("F"), Area Under the ROC Curve ("AUC"), together with macro ("M"), micro ("μ").

The rest of the chapter is organized as follows, Section 2.2 explains the different ways in which the ADR extraction task was defined. Section 2.3 presents classification approaches employed to infer the ADR extraction models. Section 2.4 explores a key-issue in machine learning, that is, the set of features used for the ADR characterization. Section 2.5 mentions alternative textual genres of corpora and languages employed in ADR extraction. Section 2.6 shows alternative assessment approaches employed to evaluate the ADR extraction approaches. That is, each factor is developed in one of the sections. Furthermore, after having reviewed a key-factor, we discuss our task and state the strategy adopted in our work. Finally, Section 2.7 provides some concluding remarks.

2.2 Definition of ADR extraction

One of the differentiating factors in related works dealing with ADR extraction is the definition of the task itself. We distinguished three definitions: presence of ADRs, ADR mentions and ADR relations (referred to as "P", "M" and "R" respectively in Table 2.1). Hereafter, we elaborate on each definition.

- **Presence of ADRs**: Some authors refer to ADR extraction as the detection of presence (or absence) of ADRs in a document. It involves a binary classification of the document itself. The classifier determines whether or not a document, such as a health record, contains an ADR (Botsis et al., 2011; Karlsson et al., 2013; Zhao et al., 2014, 2015; Friedrich and Dalianis, 2015; Henriksson et al., 2015b). This binary classification can be targeted at the entire document, as in the aforementioned cases, or at smaller parts of the document, such as at a paragraph or at a single sentence (Sohn et al., 2011; Gurulingappa et al., 2011; Patki et al., 2014; Ginn et al., 2014; Sarker and Gonzalez, 2015; Zhang et al., 2016; Huynh et al., 2016; Lee et al., 2017; Akhtyamova et al., 2017; Masino et al., 2018).

- **ADR mentions**: Other authors refer to ADR extraction as the extraction of a subset of entities, namely, the adverse reactions. These entities are those labeled as "Adverse reaction to drug" in the Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT) hierarchy. In this case, the focus is only the disease (finding, sign or symptom) resulting from some medication intake but the medication is not marked (Nikfarjam et al., 2015; Lin et al., 2015; Jagannatha and Yu, 2016a,b; Stanovsky et al., 2017; Tutubalina and Nikolenko, 2017; Cocos et al., 2017; Gupta et al., 2018; Wunnava et al., 2018). The core approach for this task is, generally, a Named Entity Recognition (NER) system particularly suited for diseases that correspond to adverse reactions (a subset of diseases in SNOMED CT).

- **ADR relations**: Others interpret the task as the extraction of relations between drugs and diseases (the adverse reactions). From the NLP perspective this consists in a relation extraction task of cause-effect events in which the drug is the causing agent and the disease the reaction or effect. With this definition, it is the related drug-disease pair itself that is referred to as ADR (Aramaki et al., 2010; Miura

et al., 2010; Gurulingappa et al., 2012a; Li et al., 2015; Henriksson et al., 2015a; Luo, 2017; Raj et al., 2017; Legrand et al., 2018; He et al., 2019). Note that this task is more thorough than the detection of presence of ADRs, since the detection of mere presence does not explicitly indicate which was the ADR). Besides, it is more comprehensive than the extraction of just the adverse reaction (the disease) as it omits the causative drug.

Positioning our work with respect to related works. The definition of ADR extraction constraints, considerably, the amount of information provided. Our work is framed within the DETEAMI and PROSAMED projects (see Section 1.2). In order to meet the needs of these projects, the ADR extraction was defined as a relation extraction task. Accordingly, we extract drug-disease pairs in which the drug is responsible of the disease considered an adverse reaction. However, this approach is more complex than the other two because it entails the recognition of the entities and the detection of causal relations. To cope with relation extraction we followed the pipeline approach shown in Figure 1.2. The majority of our work rests on the second step, that is, the detection of causal relations. To this end, as was done by Aramaki et al. (2010) and Miura et al. (2010), the gold mentions (entities manually annotated by the experts) were employed in Chapter 4, Chapter 5 and Chapter 6. Nevertheless, we completed the pipeline and automatically recognized medical entities to later extract ADRs in Chapter 7.

2.3 ADR classification techniques

Reviewing the methods employed to extract ADRs, our impression is that the mainstream follows supervised classification techniques. In this line, we distinguished the so called traditional machine learning algorithms and emerging deep learning approaches (referred to as "T" and "DL" respectively in Table 2.1).

- **Traditional machine learning algorithms**: These are the classifiers typically used to learn from the data and make a prediction employing hand-crafted features (Goldberg and Hirst, 2017). In works related with ADR extraction we can highlight classifiers such as Naive Bayes (NB), Support Vector Machines (SVM), Random Forest (RF), Decision

Trees (DT), Conditional Random Fields (CRF) or Maximum Entropy (ME).

- NB: It predicts the class based on a set of given features, assuming that features are independent given the class. It was tested in several works such as (Botsis et al., 2011; Ginn et al., 2014; Zhao et al., 2014, 2015; Sarker and Gonzalez, 2015).
- *SVM*: It selects the hyperplane that separates the feature space by their class with more confidence (Aramaki et al., 2010; Miura et al., 2010; Gurulingappa et al., 2012a). SVM was the best performing one among several classifiers such as NB, RF, DT and ME in (Patki et al., 2014; Ginn et al., 2014; Friedrich and Dalianis, 2015; Sarker and Gonzalez, 2015).
- *RF*: It is an ensemble of decision trees, where each decision tree is created with a subset of the features used to classify a given example (Henriksson et al., 2015a,b). RF outperformed other classifiers such as NB, SVM and DT in (Karlsson et al., 2013; Zhao et al., 2014, 2015).
- *DT*: It consists of a series of nodes sorted by their relevance, where each node represents a feature, the branches represents the values of them and the leaf nodes are the classes (Celli, 2010; Sohn et al., 2011).
- *CRF*: It predicts sequences of labels for sequences of input samples (Nikfarjam et al., 2015; Lin et al., 2015). It was employed to find ADR mentions.
- *ME*: It selects the class that has the largest entropy based on the principle that the distribution should be as uniform as possible, that is, has maximal entropy (Zhang et al., 2016). ME was the best performing classifier in comparison with others such as NB, SVM and DT in (Gurulingappa et al., 2011).
- *Others*: Other classifiers used for the task were structured perceptron for training and multiple-beam search algorithm for decoding (Li et al., 2015) and Generalized Additive Model (Botsis et al., 2011).

- **Deep learning algorithms**: These are neural networks with several hidden layers that not only predict, but also represent the data by

automatically inferred features (Goldberg and Hirst, 2017). We can distinguish two main neural network architectures, the Convolutional Neural Network (CNN) and the Recurrent Neural Network (RNN) and, in turn, in the last group it is possible to distinguish Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU). We can also find the Bidirectional Recurrent Neural Network (Bi-RNN), which includes the Bidirectional Long Short-Term Memory (Bi-LSTM). The ADR extraction in related works was based mainly on the aforementioned neural networks.

- *CNN*: It produces a fixed size vector representation that captures the most informative local aspects (Zeng et al., 2014; Nguyen and Grishman, 2015; Lee et al., 2017; Akhtyamova et al., 2017; Masino et al., 2018; He et al., 2019). CNN outperformed Recurrent Convolutional Neural Network, Convolutional Recurrent Neural Network and Convolutional Neural Network with Attention in (Huynh et al., 2016).
- *RNN*: It has recurrent hidden states in a way that the activation of a hidden state depends on the previous hidden state (Ebrahimi and Dou, 2015; Cocos et al., 2017).
- *LSTM*: It is a type of RNN with a gating mechanism that preserves the memory and the error gradients across time (Luo, 2017). The models inferred with LSTM performed better than those inferred with CNN in (Zheng et al., 2016; Fabregat et al., 2018; Legrand et al., 2018).
- *GRU*: It is also an RNN with a gating mechanism, but with substantially fewer gates and without a separate memory component. GRU outperformed Bi-LSTM in (Jagannatha and Yu, 2016a; Tutubalina and Nikolenko, 2017).
- *Bi-LSTM*: It is an LSTM that considers the forward and backward states (Jagannatha and Yu, 2016b; Miwa and Bansal, 2016; Zhou et al., 2016; Stanovsky et al., 2017; Li et al., 2017; Gupta et al., 2018; Wunnava et al., 2018; Christopoulou et al., 2018).
- *Others*: Some of the aforementioned classifiers were combined such as Convolutional Recurrent Neural Network (Raj et al., 2017), multichanel Bi-LSTM with CNN (Le et al., 2018) or even a CNN model for the text descriptions together with a CNN model and a

Bidirectional Recurrent Convolutional Neural Network model for the sentence representation (Ren et al., 2018).

Some of the aforementioned works compared their neural network implementations with the traditional classifiers, concluding that the neural networks offered better results with less feature engineering. This makes their use promising to improve the performance of our work. The majority of the related works dealing with ADR extraction employed supervised methods and this is what we did in this work, this is why we do not include unsupervised methods in Table 2.1. However, unsupervised methods can also be used. For example, Duque et al. (2015) created a knowledge representation model that assigns relations according to the statistical significance of their co-occurrence in the same document.

Positioning our work with respect to related works. ADR extraction evolved from traditional to deep-learning classifiers. In our work, we also explored the two types of machine learning algorithms. Specifically, in Chapter 4 and Chapter 5 we explain the experimentation made with traditional classifiers and in Chapter 6 the experimentation made with deep learning classifiers. Although in some of the works based on relation extraction the classifiers were used to jointly extract the entities and the relations, such as was done by Li et al. (2015), Miwa and Bansal (2016), Li et al. (2017) and Katiyar and Cardie (2017), we used a pipeline method.

2.4 ADR characterization features

To infer the models with the machine learning algorithms, it is necessary to represent the instances with a set of features. In any of the classification techniques the characterization of the instances plays an important role. In the classifiers mentioned previously, we distinguish symbolic and dense features for the characterization (referred to as "S" and "D" respectively in Table 2.1).

- **Symbolic features**: These features are used for the traditional machine learning algorithms employed in the most earlier related works and are mainly based on word-forms, n-grams, lemmas, etc. Often, feature engineering led the authors to incorporate other numeric features. In what follows we review the most relevant ones. It is possible to distinguish representations that contain general features.

- *Entity*: Word-forms of the entities (e.g. drug and disease) (Aramaki et al., 2010; Li et al., 2015).
- *Context*: Words between the entities (Aramaki et al., 2010; Li et al., 2015; Celli, 2010) or before and after the entities (Li et al., 2015).
- *Distance*: Number of words between the entities (Miura et al., 2010; Li et al., 2015; Celli, 2010) or surrounding the entities (Aramaki et al., 2010; Miura et al., 2010).
- *Order*: Order in which the entities involved in a pair appear (e.g. disease-drug or drug-disease) (Miura et al., 2010).
- *Lemma*: Lemmas of the corresponding word-forms (Botsis et al., 2011; Gurulingappa et al., 2011, 2012a).
- *Part-Of-Speech (POS)*: POS tags of the corresponding word-form (Gurulingappa et al., 2012a; Li et al., 2015).
- *Affixes*: Suffixes and prefixes of the corresponding word-forms (Gurulingappa et al., 2011; Li et al., 2015).
- *N-grams*: Sequence of contiguous n word-forms (Patki et al., 2014; Sarker and Gonzalez, 2015).
- *Synset expansions*: Synonyms according to WordNet (Patki et al., 2014; Sarker and Gonzalez, 2015).
- *Polarity*: Value indicating the polarity by how a change happens; if a bad thing was reduced, the outcome is positive and if a bad thing was increased, the outcome is negative (e.g. more-good, more-bad, less-good and less-bad) (Patki et al., 2014; Sarker and Gonzalez, 2015).
- *Sentiword scores*: Score that represents the general sentiment (Patki et al., 2014; Sarker and Gonzalez, 2015).
- *Others*: Words of a text segment (e.g. sentence, document) (Gurulingappa et al., 2011, 2012a; Friedrich and Dalianis, 2015) and vector where the value of the ith feature is equal to the number of times that feature or word occurs (Ginn et al., 2014).

There are also features related with the medical domain.

  - *Presence*: Value indicating the presence of side effect keywords (Sohn et al., 2011) or the presence of drugs (Karlsson et al., 2013) and number of tokens matching with the DrugBank and MedDRA lexicons (Gurulingappa et al., 2011).
  - *Clinical measurements*: Values of the measurements taken during the hospitalization, such as blood pressure or pulse rate (Zhao et al., 2014).
  - *Clinical codes*: Diagnoses encoded by the International Statistical Classification of Diseases and Related Health Problems in its 10th version (ICD-10) and drugs encoded by the Anatomical Therapeutic Chemical Classification System (ATC) (Zhao et al., 2014, 2015).
  - *Unified Medical Language System (UMLS) medical semantic types and Concept Unique Identifiers (CUIs)*: Frequency of these terms in a document (Sarker and Gonzalez, 2015).
  - *ADR lexicon*: Matches with a lexicon formed with ADR mentions (Sarker and Gonzalez, 2015).

- **Dense features**: These features consist of n-dimensional ($\mathbb{R}^n$) vectors created from embeddings (vectorial representation of the words generated with unsupervised methods). They can be embeddings or features derived from them. On the one hand, we can differentiate the dense representations used for the traditional classifiers. Sometimes these dense features are combined with the symbolic ones.

  - *Vectors*: Sum of the semantic vectors of each word to represent the context (Henriksson et al., 2015a), concatenation of vectors from semantic spaces built with different context window sizes (Henriksson et al., 2015b) and average of vectors of words (Zhang et al., 2016).
  - *Clusters*: Number of the cluster associated to the token, obtained by K-means clustering on the word-embeddings (Nikfarjam et al., 2015; Lin et al., 2015).

  On the other hand, we can differentiate the dense features employed by the neural networks, which are automatically inferred from dense core-features.

- *Word embedding*: Embedding corresponding to the word (Huynh et al., 2016; Jagannatha and Yu, 2016a,b; Zheng et al., 2016; Akhtyamova et al., 2017; Cocos et al., 2017; Raj et al., 2017; Katiyar and Cardie, 2017; Gupta et al., 2018; Masino et al., 2018). This can be augmented with knowledge graph embeddings of DBpedia, (Stanovsky et al., 2017) or with the concatenation of the character-level representation (Tutubalina and Nikolenko, 2017; Wunnava et al., 2018). This feature apart from being used alone, it was also combined with other core-features described below (Zeng et al., 2014; Ebrahimi and Dou, 2015; Nguyen and Grishman, 2015; Miwa and Bansal, 2016; Zhou et al., 2016; Katiyar and Cardie, 2017; Christopoulou et al., 2018; Ren et al., 2018; Le et al., 2018).
- *Position embeddings*: Embeddings of the relative distances of the current word to the entities involved in a relation (Zeng et al., 2014; Nguyen and Grishman, 2015; Zhou et al., 2016; Luo, 2017; Christopoulou et al., 2018; Ren et al., 2018).
- *Dependency type embedding*: Embedding of the dependency type with the parent in the dependency tree (Miwa and Bansal, 2016; Li et al., 2017).
- *Entity embeddings*: Embeddings of the entities involved in a relation (Li et al., 2017; Legrand et al., 2018).
- *POS embedding*: Embedding corresponding to the POS tag of a word (Miwa and Bansal, 2016; Le et al., 2018).
- *Entity type embedding*: Embedding corresponding to the semantic type of the entity (Miwa and Bansal, 2016; Christopoulou et al., 2018).
- *WordNet embeddings*: One-hot vectors that determine which WordNet super-senses the token belongs to (Le et al., 2018).
- *Character embedding*: Character-level embedding corresponding to each character of a token (Le et al., 2018).
- *Others*: Concatenation of the concept position, the concept contents, the concept types and the relation type (He et al., 2019) and different phrase embeddings (Lee et al., 2017).

Apart from the features associated with the entities and their context, the features related with the distance and their position seem particularly interesting since we represent the ADRs as drug-disease pairs.

Positioning our work with respect to related works. In our work we explored both symbolic and dense representations. The symbolic features chosen for the traditional classifiers are explained in Chapter 4, the dense features for the traditional classifiers are explained in Chapter 5 and the dense features for the deep learning algorithms are explained in Chapter 6. In the related works we observed that, initially, a wide variety of features were proposed and, in some cases, the authors selected a subset of relevant features. In our case, we did the same for both symbolic and dense features. Furthermore, for the dense representations the authors employed either medical embeddings or generic embeddings. In our case, we explored both of them, but we focused on medical embeddings (particularly, for the initialization of the core-features of the deep learning algorithms). Note that the core-features can be initialized randomly or with pre-trained embeddings.

2.5 Corpora for ADR extraction

ADR extraction was applied to several textual genres. Furthermore, most of the available corpora for this task are written in English and developing approaches for languages other than English is challenging (Dalianis, 2018).

Regarding the ***textual genres***, we found three types in the related works: EHRs, text from social media and scientific publications (referred to as "E", "SM" and "SC" respectively in Table 2.1).

- **EHRs**: They are written by experts on the medical domain, they do not use a fully formal register and they can contain abbreviations or typos. We found a few works dealing with EHRs (Aramaki et al., 2010; Miura et al., 2010; Sohn et al., 2011; Karlsson et al., 2013; Zhao et al., 2014, 2015; Friedrich and Dalianis, 2015; Henriksson et al., 2015a,b; Jagannatha and Yu, 2016a,b; Luo, 2017; Raj et al., 2017; Wunnava et al., 2018; He et al., 2019).

- **Social media**: These corpora mainly consist of comments from social networks such as Twitter or medical forums, which are written by non-experts using a colloquial register with abbreviations and often contain typos. There are several corpora that comprise texts from social media (Patki et al., 2014; Ginn et al., 2014; Sarker and Gonzalez, 2015; Nikfarjam et al., 2015; Lin et al., 2015; Zhang et al., 2016; Huynh et al., 2016; Stanovsky et al., 2017; Lee et al., 2017; Tutubalina and

Nikolenko, 2017; Akhtyamova et al., 2017; Cocos et al., 2017; Gupta et al., 2018; Masino et al., 2018).

- **Scientific publications**: These corpora are written by experts using a formal register and seldom have typos. There are several corpora that comprise texts from scientific publications (Gurulingappa et al., 2011, 2012a; Li et al., 2015; Sarker and Gonzalez, 2015; Huynh et al., 2016; Li et al., 2017; Legrand et al., 2018).

In some of the aforementioned works, several corpora of different textual genres were used (Li et al., 2017; Raj et al., 2017; Legrand et al., 2018).

With regard to the ***language***, in the medical domain there is interest in working with languages other than English. In fact, we found clinical corpora written in different languages, some of which are mentioned briefly below. After that, we focus on the corpora shown in works related with ADRs, paying special attention to those written in Spanish.

There are clinical corpora written in other languages that do not have ADRs annotated explicitly. For French, there are a corpus that comprises case reports and EHRs (Deléger et al., 2014) and the Medical Entity and Relation LIMSI annOtated Text (MERLOT) corpus (Campillos et al., 2018), which comprises clinical notes from different hospitals. For German, there is a corpus that consists of discharge summaries and clinical notes (Roller et al., 2016) and a corpus that comprises EHRs (Zubke, 2017). For Italian, there is a corpus formed by medical reports, including discharge summaries, diagnoses and medical test reports (Attardi et al., 2015). For Portuguese, there is a corpus with electronic medical records from different specialties (Lamy et al., 2018). In all these corpora the entities and relations between them were annotated, except in (Zubke, 2017), with annotations about the numeric values, and in (Attardi et al., 2015), with annotations about medical entities and their relations with the negation and speculation.

The same happens with other clinical corpora written in Spanish. The UHU-HUVR corpus (Cruz et al., 2017) and the IULA Spanish Clinical Record Corpus (IULA-SCRC) (Marimon et al., 2017) consist of clinical reports with annotations about negation and are publicly available. The Spanish corpus extracted from the MultiMedica corpus (Moreno-Sandoval and Campillos-Llanos, 2013) contains journalistic texts from OCU-Salud as well as encyclopedic articles from Tu otro médico and was tagged with POS (Llanos and Ueda, 2015). The eHealth-KD corpus (Piad-Morffis et al., 2019) has articles

collected from MedlinePlus and was annotated with semantic concepts and relations. The DrugSemantics corpus (Moreno et al., 2017) consists of texts from Summaries Product Characteristics, with annotations about entities such as drugs or diseases (the adverse effects are included in the diseases). The BARR2 corpus (Intxaurrondo et al., 2018) has documents from Scientific Electronic Library Online (SciELO), which contains scientific journals of Latin America, South Africa and Spain and were annotated with medical abbreviations.

Hereafter, we present the corpora explored in related works for ADR extraction sorted by language. These languages are English, Japanese and Swedish (referred to as "EN", "J" and "SW" respectively in Table 2.1). Moreover, we present a corpus written in Spanish with annotations about ADRs.

- **English**
  - Corpus of 237 electronic medical records from patients in the psychiatry and psychology department at Mayo Clinic. The records were annotated with the relations between the side effect and the causative drug. This corpus was used by Sohn et al. (2011).
  - EHRs from cancer patients such as 780 EHRs used by Jagannatha and Yu (2016a), 1,154 EHRs used by Jagannatha and Yu (2016b) or 1,089 EHRs from 21 cancer patients provided by University of Massachusetts for the challenge for Detecting Medication and Adverse Drug Events from Electronic Health Records used by (Wunnava et al., 2018). This last set of EHRs was released to the participant. In all the aforementioned records several medical entities were annotated, including adverse drug event mentions.
  - Corpus of the 2010 i2b2/VA relation challenge (Uzuner et al., 2011), which consists of 871 discharge summaries and progress reports provided by Partners Healthcare, Beth Israel Deaconess Medical Center and the University of Pittsburgh Medical Center. The documents were annotated with different relations among medical problems, tests and treatments, such as medical problem caused by a treatment. This corpus is publicly available and was used by Luo (2017), Raj et al. (2017) and He et al. (2019).
  - Corpus created through the extraction of tweets related to 74 drugs of interest from Twitter (Ginn et al., 2014), which contains

a total of 10,822 tweets. They were labeled with medical mentions that include the adverse drug reactions. This corpus was employed in the Pacific Symposium on Biocomputing 2016 social media mining shared task for ADR classification (Sarker et al., 2016) and was used by Ginn et al. (2014), Lin et al. (2015), Zhang et al. (2016), Lee et al. (2017), Akhtyamova et al. (2017) and Masino et al. (2018).

- Other corpus created through the extraction of tweets related to 81 drugs of interest (Nikfarjam et al., 2015), which contains 1,784 tweets. They were labeled with medical mentions that include the adverse drug reactions. This was used by Nikfarjam et al. (2015), Cocos et al. (2017) and Gupta et al. (2018).
- Corpus created with 10,617 comments from DailyStrength, a health related social network where people share their personal knowledge and experiences regarding diseases and/or treatments (Patki et al., 2014). This corpus was also labeled with medical mentions, including adverse drug reactions. It was used by Patki et al. (2014).
- Other corpus created with 6,279 comments from DailyStrength (Nikfarjam et al., 2015). These comments were also labeled with medical mentions, including adverse drug reactions. It was used by Nikfarjam et al. (2015).
- CSIRO Adverse Drug Event Corpus (CADEC) (Karimi et al., 2015), which contains 1,250 posts from Ask a Patient, a medical forum that collects ratings and reviews of medications from their consumers. This corpus was labeled with medical entities, including adverse drug reaction mentions. It was used by Tutubalina and Nikolenko (2017) and Stanovsky et al. (2017).
- ADE corpus (Gurulingappa et al., 2012b), which contains 2,972 PubMed case reports. They were labeled with relationships between the drugs and adverse effects as well as between the drugs and dosages. It was used by Gurulingappa et al. (2011), Gurulingappa et al. (2012a), Li et al. (2015), Li et al. (2017) and Legrand et al. (2018).
- EU-ADR corpus (Van Mulligen et al., 2012), which consist of 100 Medline abstracts obtained from PubMed. The drug, disease, tar-

gets and their relationships were annotated (the diseases correspond to ADR mentions). It was used by Legrand et al. (2018).

- Corpus of 6,034 case reports that were submitted to Vaccine Adverse Event Reporting System. These reports were labeld as positive or negative according to the presence of adverse events. This corpus was used by Botsis et al. (2011).

- **Swedish**

  - Dataset extracted from the Stockholm EPR Corpus (Dalianis et al., 2012), comprising 700,000 health records from Karolinska University Hospital in Stockholm. This corpus was annotated with medical entities and relations such as adverse events. It was used by Karlsson et al. (2013), Zhao et al. (2014), Zhao et al. (2015), Friedrich and Dalianis (2015), Henriksson et al. (2015a) and Henriksson et al. (2015b).

- **Japanese**

  - Corpus that comprises 3,012 discharge summaries gathered from all departments of the University of Tokyo Hospital. It was annotated with adverse effect relations between symptoms and drugs and was use by Aramaki et al. (2010) and Miura et al. (2010).

- **Spanish**

  For Spanish, we found a corpus that, as explained below, was used with interesting techniques based on rules and unsupervised methods. In this case, we focus on supervised machine learning algorithms for relation extraction, this is why Table 2.1 just includes works that employed them.

  - SpanishADRCorpus (Segura-Bedmar et al., 2014b) labeled with drugs and effects as entities and drug indications and adverse drug reactions as relations. It is composed by 400 documents gathered from ForumClinic, a health network website in Spanish. Note that, this corpus was employed for ADR extraction with techniques based on rules and unsupervised methods. For example, Segura-Bedmar et al. (2014b) identified drug and adverse event mentions by a dictionary-matching approach. Instead, we

approached this task as relation extraction. Segura-Bedmar et al. (2014a) and de la Peña et al. (2014) identified drugs and effects also with a dictionary-based approach and next, extracted indication and ADR relations with a method based on co-occurrence. In this method, the pairs co-occurring within a window of n tokens are treated as relations and the indication and ADR relations are found by looking-up a table created previously with these relations. Segura-Bedmar et al. (2015) identified drugs and effects with another dictionary-based approach and next, extracted indication and ADR relations using distant-supervision. With the distant supervision approach, two entities that co-occur in a sentence form a relation and a knowledge base is used instead of an annotated corpus for the learning process.

Positioning our work with respect to related works. In our work we focus on EHRs written in Spanish. Obtaining large sets of EHRs is difficult, even more if they are written in languages other than English, since they contain personal information of the patients that cannot be publicly available. Most often, the authors give the number of documents of the corpus. Nevertheless, the average number of tokens per document is missed and this makes difficult to make a comparison. For example, in other works they compare corpus of EHRs with a very different number of documents (over 100,000 for Spanish and over 1,000,000 for Swedish) but a similar number of tokens (about 50 millions) (Pérez et al., 2017). In our case, we tried to give an exhaustive analysis of our corpora in Chapter 3.

2.6 Evaluation of ADR extraction

The related works assessed their experiments in different ways. These assessments may differ in terms of evaluation scheme as well as evaluation metrics. In Table 2.1, we can see the results obtained in these experiments together with the corresponding evaluation schemes and metrics.

With regard to the ***evaluation schemes***, we found that the majority of the works reported a **hold-out** evaluation scheme, even though, **k-fold cross-validation** is also popular (referred to as "HO" and "CV" respectively in Table 2.1). In other cases the authors used both of them depending on the experiment.

Regarding the ***evaluation metrics***, the majority of the works reported the **precision**, **recall** and **f-measure**. Some exceptions are the **Area Under ROC Curve (AUC)** showing or not the **Receiver Operating Characteristic (ROC) curve** and the **accuracy**. Note that sometimes, the authors only offered the averaged results of all the classes. For example, the macro-averaged measures or the micro-averaged measures. In this case, we show the f-measure of the positive class whenever it is possible (referred to as "F" in Table 2.1).

Positioning our work with respect to related works. In our work we employed hold-out and k-fold cross-validation. We also employed several evaluation metrics, paying particular attention to the f-measure of the positive class. The reason is that the bias in the learning, due to the class imbalance, makes easier to predict correctly the majority class than the minority class and, hence, we prefer to study the difficult situation (the f-measure of the positive class). The evaluation schemes and metrics used to assess our experiments are summarized in Section 3.3.

2.7 Concluding remarks

In this chapter we shown the main differences found in the related works to tackle the ADR extraction. The most outstanding differentiating factor rest on the definition of ADR extraction and, accordingly, the NLP approach adopted to tackle it. ADR extraction defined as relation extraction is the most comprehensive one since it clearly reveals both the causative drug and the caused disease. In this work we pay particularly attention to the ADR discovery step. To deal with this, there are numerous techniques, in our case, we opted for supervised classification. We developed traditional approaches as well as emerging approaches based on deep-learning. The main difference rests on the features employed, either hand-crafted or automatically inferred. As in related works, we found the characterization of ADRs a key issue and we explored both symbolic and dense features.

In addition, related works developed their experiments with corpora of different textual genres and mainly in English. In our case, we focus on EHRs written in Spanish. Apart from the corpora, the evaluation schemes and metrics also differ in related works. This makes difficult to compare the performance of the approaches proposed throughout all the works.

3 Experimental framework

3.1 Introduction

Before starting with ADR extraction, we find necessary to explain some details of the experimental framework. This is an attempt to help the reader to understand the results of the experiments developed in the next chapters. Specifically, the aim is to explain the main characteristics of the corpus used in this work and describe the evaluation process carried out to asses the models inferred from the corpus.

The rest of the chapter is organized as follows: Section 3.2 describes the different corpora used during this work, the annotated as well as the unannotated ones. Section 3.3 explains the evaluation schemes and the evaluation metrics used to evaluate the predictive models. Section 3.4 provides the concluding remarks.

3.2 Corpora

This work was developed using EHRs, which are defined as 'a repository of information regarding the health of a subject of care, in computer processable form' (ISO, 2005). The EHRs are often written by the doctors during the actual consultation. Thus, these records present some challenges in relation to i) confidentiality, ii) structure, and iii) lexical variability.

Regarding **confidentiality**, the EHRs contain sensitive information about the patients (Cohen and Demner-Fushman, 2014), which makes them diffi-

cult to access. In our case, in order to cope with legal and ethical issues, we worked with documents that were dissociated in advance by the Basque Health Service (Osakidetza), according to the DETEAMI and PROSAMED projects (see Section 1.2). In this way, the EHRs did not contain any personal information regarding the patients. In addition, the EHRs used in this work were subject to an agreement between Osakidetza and the University of the Basque Country.

Regarding **structure**, the EHRs are written in a free style, that is to say, they are not structured in standardized sections (Cohen and Demner-Fushman, 2014). Specifically, these EHRs are semi-structured, which means that they have two main fields. The first one for personal data of the patient (name, age, dates relating to admittance, etc.) that were not provided by the hospital for privacy issues. The second one contains the antecedents, clinical analysis, evolution, diagnosis, treatment, etc. This second field is unstructured in our corpus, although some other hospitals use templates that divide this into several sub-fields.

Regarding **lexical variability**, the EHRs frequently contain standard and non-standard abbreviations, misspellings or punctuation errors because they are written under time pressure (Leaman et al., 2015; Dalianis, 2018). These variants in the terminology make these records difficult to process using general-purpose NLP tools (Dalianis, 2018).

Figure 3.1 shows some examples of EHRs. In Figure 3.1a we can observe that the entire EHR does not contain personal information about the patient or the doctor because it was dissociated previously. We can also observe that Figure 3.1a and Figure 3.1b do not follow the same structure: in Figure 3.1a we can distinguish only the sections '*diagnostico*' (meaning 'diagnosis') and '*tratamiento*' (meaning 'treatment'), whereas in Figure 3.1b we can distinguish the sections '*evolución*' (meaning 'evolution') and '*impresión diagnostica*' (meaning 'diagnostic impression'). Furthermore, in both cases we can see abbreviations such as '*I.V.*' (meaning 'intravenous'), '*IZQ.*' (meaning 'left'), '*gr.*' (meaning 'gram') or '*ADH*' (meaning 'antidiuretic hormone') and misspellings and punctuation errors such as '*diagnostico*' instead of '*diagnóstico*' (meaning 'diagnostic'), '*flemon*' instead of '*flemón*' (meaning 'abscess') or '*estadío*' instead of '*estadio*' (meaning 'stage').

```
Paciente que ingresa por presentar odinofagia.
Habiéndosele efectuado tratamiento médico I.V., es dado de alta en
el día de la fecha.
DIAGNOSTICO: - FLEMON PERIAMIGDALINO IZQ.
TRATAMIENTO:
- AUGMENTINE 850:  1/8 horas (1 semana).
- IBUPROFENO 600:  1/8 horas (3 días), posteriormente, si dolor.
- PARACETAMOL 1 gr.:  1/6 horas, si más dolor.
Seguirá control por su médico de atención primaria.
```

(a) Entire short EHR.

```
EVOLUCIÓN:
El paciente presenta una enfermedad de Parkinson idiopática de
predominio rígido-acinética con importante rigidez axial y de
predominio derecho (estadío 3 de Hoehn-Yahr).
Asimismo, presenta una hiponatremia atribuible a baja ingesta de
sodio y toma de tiazídicos bien tolerada.
Posteriormente, ha sido valorado por Nefrología quienes consideran
probable que asocie una secreción inadecuada de ADH a las
circunstancias previamente citadas, aconsejando restricción hídrica
y dieta salada.
No se han apreciado otras alteraciones agudas sobreañadidas.
Se han suspendido los tiazídicos y se eleva discretamente el
sinemet.
Asimismo, ha presentado hipoglucemias matutinas asintomáticas, por
lo que se disminuye la dosis de metformina.
IMPRESIÓN DIAGNÓSTICA:
- ENFERMEDAD DE PARKINSON IDIOPÁTICA ESTADÍO 3-4 DE HOEHN YAHR
- HIPONATREMIA NORMOVOLÉMICA EN RELACIÓN CON BAJA INGESTA DE
SODIO, TIAZÍDICOS, DIARREA Y PROBABLEMENTE UN SÍNDROME DE SECRECIÓN
INADECUADA DE ADH SOBREAÑADIDO
- LOS ANTERIORES
```

(b) Extract from large EHR.

Figure 3.1: Examples of EHRs in raw text.

3.2.1 Annotated corpora

Manually annotated EHRs were used to infer the ADR extraction model and also to assess its predictions with respect to the ground truth. If the access to EHRs was difficult, the access to annotated EHRs is even more difficult since the annotation process require experts and is very time-consuming. Figure 3.2 shows the EHRs of the examples given previously in Figure 3.1 with the corresponding annotations of the medical entities and their relationships. Note that the terms that correspond to medical entities can be monolexical and polilexical or sintagmatic. The monolexical terms are formed by one word, for example, '*hiponatremia*' (meaning 'hyponatremia'). The polilexical or syntagmatic terms are formed by more than one word, for example, '*enfermedad de Parkinson idiopática*' (meaning 'idiopathic Parkinson's disease'). The documents were annotated with the annotation toolkit Brat (Stenetorp et al., 2012) and following mainly the process explained by Oronoz et al. (2015).

Among the **entities** labeled in the EHRs we can distinguish:

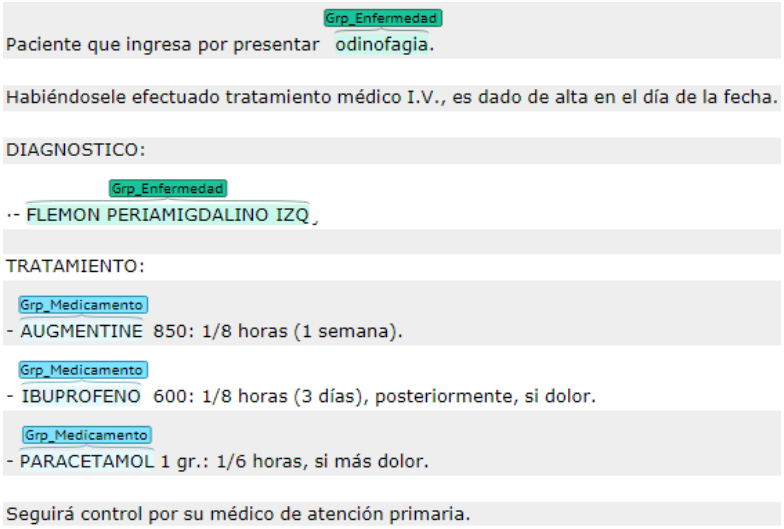
i Diseases, signs and symptoms joined in the Disease group and denoted as "Grp_Enfermedad". For example, in Figure 3.3 we found '*HTA*', '*HDA*', '*ulcus gástrico*', '*hepatopatía crónica*', '*cirrosis*', '*hipertensión portal*', '*varices esofágicas*', '*bradicardia sinusal*', '*descompensaciones ascíticas*' (meaning 'HBP', 'upper GI bleeding', 'gastric ulcer', 'chronic liver', 'cirrhosis', 'portal hypertension', 'esophageal varices', 'sinus bradycardia', 'ascitic decompositions' respectively).

ii Allergies denoted as "Alergia". For example, in Figure 3.3 we found '*alergias medicamentosas*' (meaning 'drug allergies').

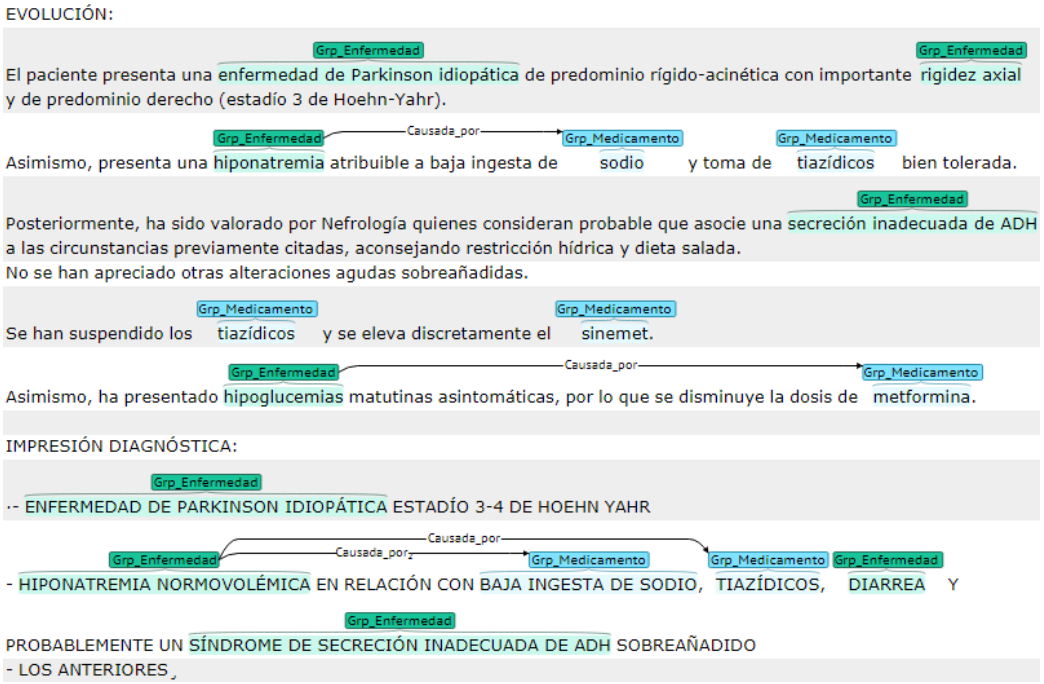
iii Brand-name drugs, substances and active principles joined in the Drug group and denoted as "Grp_Medicamento". For example, in Figure 3.3 we found '*Aines*', '*etílica*', '*betabloqueante*' (meaning 'NSAIDs', 'ethylic', 'beta-blocker' respectively).

iv Procedures denoted as "Procedimiento". For example, in Figure 3.3 we found '*paracentesis evacuadora*' (meaning ' evacuative paracentesis').

In our work we focus on the following entities: Disease group, Allergies and Drug group.



(a) Entire short EHR manually annotated by the experts.



(b) Extract from large EHR manually annotated by the experts.

Figure 3.2: Examples of EHRs with medical entities and relations annotated by the experts.

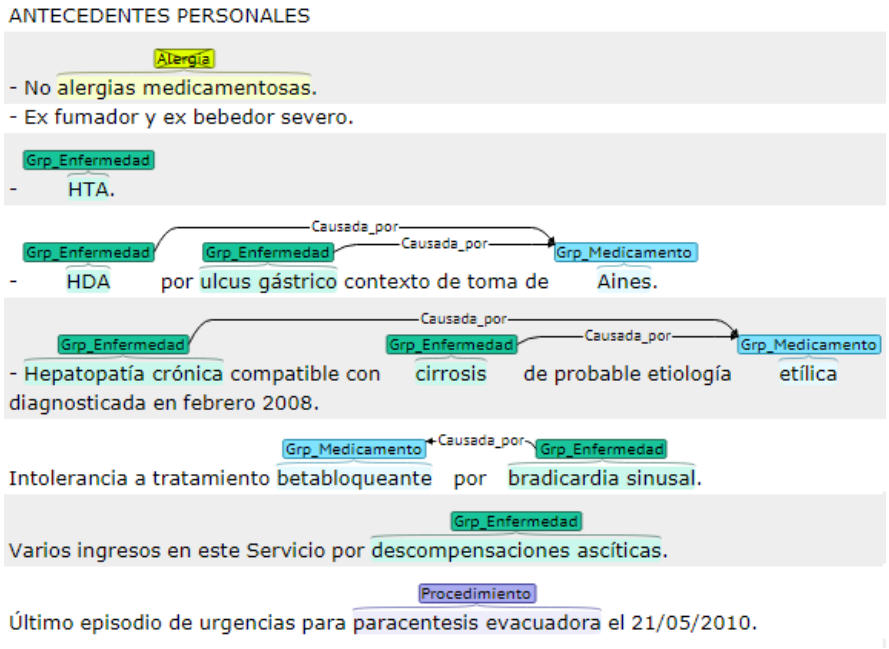
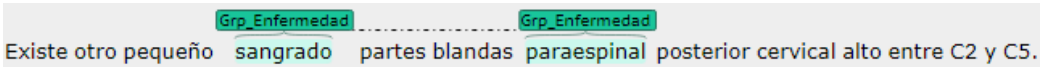


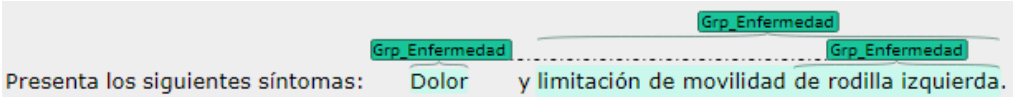
Figure 3.3: Extract from an EHR with annotations of the different types of medical entities.

There are some entities that are not a continuous sequence of words. Figure 3.4 shows examples of **discontinuous entities**. In Figure 3.4a appears the simple discontinuous entity '*sangrado paraespinal*'. In Figure 3.4b the discontinuous entity '*dolor de rodilla izquierda*' overlaps with the entity '*limitación de movilidad de rodilla izquierda*' given that there is a compound sentence.

There are also some entities that appear negated. In all the cases, the entities that appear modified by a negation cue correspond to "Grp_Enfermedad" or "Alergia". In Figure 3.5 we can see examples of **negated entities**. In Figure 3.5a the entity '*dislipemia*' is negated by the negation cue '*no*' and '*hiperuricemia*' is negated by '*ni*'. In Figure 3.5b the entity '*déficit motor*' is negated by the negation cue '*no*', the discontinuous entity '*déficit sentitivo*' is negated by the same negation cue '*no*' and the discontinuous entity '*déficit de campo visual*' is negated by the negation cue '*ni*'. In Figure 3.5c the entity '*febril*' is negated by the prefix '*a*'.

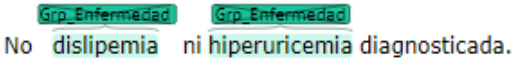


(a) Discontinuous entity without overlap.

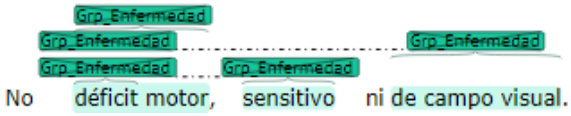


(b) Discontinuous entity with overlap.

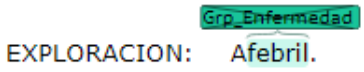
Figure 3.4: Examples of discontinuous entities. The discontinuity is represented with a discontinuous line. Figure 3.4a means 'There is other soft parts small bleeding in high posterior cervical paraspinal between C2 and C5.'. Figure 3.4b means 'He presents the following symptoms: Pain and limited mobility of left knee.'.



(a) Negated entities.



(b) Discontinuous negated entities.



(c) Negated entity with prefix.

Figure 3.5: Examples of negated entities. The negation is represented with a cross. Figure 3.5a means 'No dyslipidemia nor diagnosed hiperuricemia.'. Figure 3.5b means 'No motor, sensory nor visual field deficit.'. Figure 3.5c means 'Exploration: Afebrile.'.

Among the **relations** labeled in the EHRs we can distinguish:

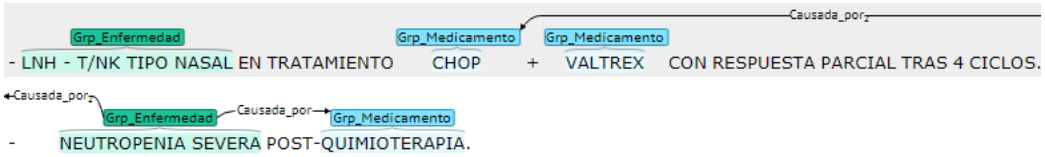
i Relations between the Disease group, the allergies and the Drug group labeled as "Causada_por" resulting in the pairs (Disease group, Drug group), (Disease group, Disease group) and (Allergy, Drug group) .

ii Relations between the Disease group and the procedures labeled as "Relacion_con" resulting in the pair (Disease group, Procedure).

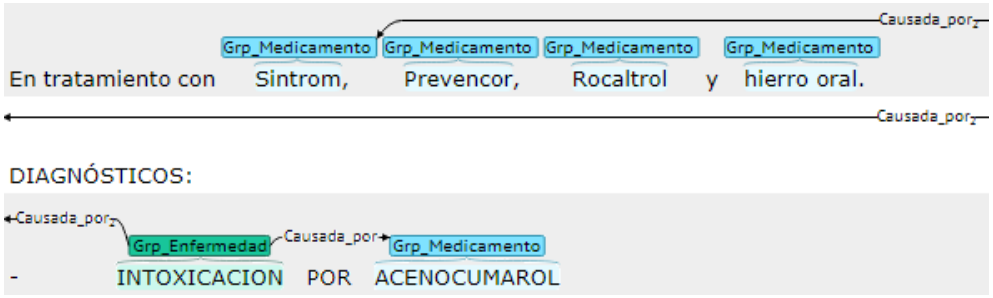
In our work we focus only on the following relations: (Disease group, Drug group) and (Allergy, Drug group).

These relations can be both inter-sentence and intra-sentence. The **intra-sentence relations** have the entities in the same sentence and the **inter-sentence relations** have the entities in different sentences. Note that we use the sentence splitting made by FreeLing-Med (Oronoz et al., 2013), which takes into account the full stops and line breaks. Figure 3.6 shows examples of ADR events labeled by the experts. In Figure 3.6a there is an intra-sentence ADR between the entities '*neutropenia severa*' and '*quimioterapia*' and an inter-sentence ADR between '*neutropenia severa*' and '*CHOP*'. In Figure 3.6b there is an intra-sentence ADR between the entities '*intoxicación*' and '*acenocumarol*' and an inter-sentence ADR between '*intoxicación*' and '*sintrom*'. Note that '*CHOP*' is an acronym for a type of chemotherapy and '*acenocumarol*' is a component of '*sintrom*'.

In this work we employed three annotated corpora that involve EHRs from two hospitals. The first one, the gold standard corpus (IxaMed-GS), was used along the main experimentation for ADR extraction in Chapter 4, Chapter 5 and Chapter 6 and also for negation detection in Appendix A and entity recognition in Appendix B. Within the framework of the DETEAMI and PROSAMED projects (see Section 1.2), more EHRs were harvested yielding other two corpora: the cross hospital corpus (IxaMed-CH), which contains the EHRs of IxaMed-GS, and the extended corpus (IxaMed-E), which contains some EHRs of IxaMed-CH but not of IxaMed-GS. Both were used for the final experimentation in Chapter 7. Note that the EHRs of the IxaMed-CH corpus and the IxaMed-E corpus were annotated during the development of the thesis, only the IxaMed-GS corpus was available from the beginning of our research work.



(a) Inter-sentence and intra-sentence ADRs.



(b) Inter-sentence and intra-sentence ADRs.

Figure 3.6: Examples of inter-sentence and intra-sentence ADRs. The ADRs are represented by an arrow that links the drug and the disease involved. Figure 3.6a means 'NHL - T/NK nasal type in treatment with CHOP + Valtrex with partial response after 4 cycles. Severe neutropenia post-chemotherapy.'. Figure 3.6b means 'In treatment with Sintrom, Prevencor, Rocaltrol and oral iron. Diagnoses: acenocumarol intoxication'.

Figure 3.7 shows graphically the distribution of the EHRs along the three corpora, which are explained more in depth in the sections below.

Figure 3.7: Venn diagram of the annotated EHRs in the gold standard corpus (denoted as IxaMed-GS), the cross hospital corpus (IxaMed-CH) and the extended corpus (IxaMed-E).

In order to infer and evaluate the models, each corpus was divided in train, development and test sets randomly selected without replacement. The division was done according to the number of documents, then the proportion between the number of positive and negative instances can be different for each set. Note that, since IxaMed-GS is part of IxaMed-CH, the sets made with the IxaMed-GS corpus were maintained when it was joined with the new documents of the IxaMed-CH corpus. Moreover, while the positive ADR relations were those manually annotated by the experts, the negative relations were created by combining all the Disease group and Allergy entities with all the Drug group entities present in each document. Next, the main characteristics of the three corpora are explained more in depth.

Gold Standard corpus

The gold standard corpus, named IxaMed-GS (Oronoz et al., 2015), consists of EHRs written in Spanish from a hospital within Osakidetza, the Galdakao-Usansolo hospital. It comprises some of the discharge records generated from 2008 to 2012, manually annotated by two experts from the pharmacy and pharmacovigilance departments of the hospital. The Inter Annotator Agreement (IAA) was 90.53% for entities and 82.86% for relations and a consensus was reached at the end.

Table 3.1 provides the quantitative description of the corpus: number of documents, word-forms, vocabulary, Out-Of-Vocabulary (OOV) words and

medical entities that the experts manually tagged in IxaMed-GS together with the number of ADR relations of each class. The OOVs are words of the evaluation set that were not seen in the training set. For example, the words '*cefaleas*', '*colecistitis*', '*cortisol*' and '*embolismo*' (meaning 'headaches', 'cholecystitis', 'cortisol' and 'embolism' respectively) appear in the evaluation set but not in the training set. This corpus has 75 EHRs that sum up to 41,633 word-forms. The vocabulary established by the train set is of size 4,934, producing 1,526 OOVs in the dev set and the vocabulary established by the train and dev sets is of size 6,460, producing 979 OOVs in the test set. Around 9% of the entities are discontinuous and about 20% of the entities are negated. 82% of the relations annotated as ADRs by the experts were intra-sentence. With respect to all candidate relations (the relations that the system explores), the intra-sentence relations represent just 1%. Note that the corpus is highly imbalanced, that is, there is a highly unequal distribution of the instances of each class. As it was expected, there are many more drug-disease pairs unrelated (class $\ominus$) than related as ADR (class $\oplus$). Specifically, in the train set there is a total of 22,459 negative relations but only 69 positive relations.

IxaMed-GS		**Train**	**Dev**	**Test**
Documents		41	17	17
Word-forms		20,689	11,246	9,698
Vocabulary		4,934	-	-
OOVs		-	1,526	979
Entities	**Drug**	503	346	354
	Disease	1,341	737	629
	Negated	399	214	150
	Non-negated	1,445	869	833
Relations	**Inter- and Intra-sentence** $\oplus$	69	45	33
	Inter- and Intra-sentence $\ominus$	22,459	17,363	24,187
	Intra-sentence $\oplus$	53	30	27
	Intra-sentence $\ominus$	231	134	173

Table 3.1: Quantitative description of the IxaMed-GS corpus. Positive relations (denoted as $\oplus$) refer to ADRs while negative relations ($\ominus$) refer to non-ADRs.

Admittedly, the number of documents might seem small. For example, in related works that find ADRs as in a binary relation classification task, the authors employed about 400 documents (Aramaki et al., 2010; Miura et al., 2010; Henriksson et al., 2015a) or even about 2,000 (Li et al., 2015, 2017). However, an striking characteristic of our EHRs is that the average number of tokens per document (500) is higher than in other EHRs, such as those used for Swedish (50) (Pérez et al., 2017). In any case, we also assessed our approach using corpora with more documents.

Cross Hospitals corpus

The cross hospital corpus, referred to as IxaMed-CH, contains EHRs from two hospitals within Osakidetza, the Galdakao-Usansolo and Basurto hospitals. The documents of the first hospital comprised some of the discharge records generated from 2008 to 2012 and the document of the second hospital comprised some of the discharge records generated from 2014. Interestingly, in these EHRs, the domain was preserved (medical domain), while the subdomains were not exactly the same, as both hospitals count on different specialties (e.g. internal medicine, cardiology, etc.). The new documents were labeled by other two expert annotators following the guideline created for the annotation of the gold standard. Then, the way to make annotations can differ in some cases from the used in the previous corpus. The IAA achieved was 89.66% for entities and 71.42% for relations, which was estimated as an extrapolation of the agreement obtained in a set of randomly selected EHRs, and a consensus was reached at the end.

Table 3.2 provides the quantitative description of the corpus. This corpus has 267 EHRs that sum up to 158,263 word-forms. The vocabulary established by the train set is of size 13,809, producing 2,628 OOVs in the dev set and the vocabulary established by the train and dev sets is of size 16,437, producing 2,280 OOVs in the test set. Around 2% of the entities are discontinuous and about 19% of the entities are negated. 83% of the relations annotated as ADRs by the experts were intra-sentence. With respect to all candidate relations, the intra-sentence relations represent just 1%. Note that the corpus is again highly imbalanced. In the train set of IxaMed-CH the proportion of negative relations is higher than the proportion of positive relations in relation to IxaMed-GS, hence, the class imbalance challenge is harder.

IxaMed-CH		**Train**	**Dev**	**Test**
Documents		157	55	55
Word-forms		91,088	34,004	33,171
Vocabulary		13,809	-	-
OOVs		-	2,628	2,280
Entities	**Drug**	2,436	943	887
	Disease	6,828	2,328	2,473
	Negated	1,716	602	662
	Non-negated	7,548	2,669	2,698
Relations	**Inter- and Intra-sentence** $\oplus$	237	96	76
	Inter- and Intra-sentence $\ominus$	132,382	46,299	53,726
	Intra-sentence $\oplus$	197	79	62
	Intra-sentence $\ominus$	2,162	366	559

Table 3.2: Quantitative description of the IxaMed-CH corpus. Positive relations (denoted as $\oplus$) refer to ADRs while negative relations ($\ominus$) refer to non-ADRs.

Extended corpus

The extended corpus, referred to as IxaMed-E, also contains EHRs from the Galdakao-Usansolo and Basurto hospitals. These EHRs were labeled in the same way that the new documents of the IxaMed-CH corpus.

Table 3.3 provides the quantitative description of the corpus. This corpus has 463 EHRs that sum up to 230,040 word-forms. The vocabulary established by the train set is of size 18,003, producing 3,182 OOVs in the dev set and the vocabulary established by the train and dev sets is of size 21,185, producing 2,735 OOVs in the test set. Around 0.3% of the entities are discontinuous and about 21% of the entities are negated. 86% of the relations annotated as ADRs by the experts were intra-sentence. With respect to all candidate relations, the intra-sentence relations represent 8%. Note that in the train set of IxaMed-E the proportion of negative relations is higher than the proportion of positive relations in relation to IxaMed-GS and IxaMed-CH, hence, the class imbalance challenge is even harder.

IxaMed-E		Train	Dev	Test
Documents		279	92	92
Word-forms		138,695	47,487	43,858
Vocabulary		18,003	-	-
OOVs		-	3,182	2,735
Entities	**Drug**	3,474	1,128	1,122
	Disease	10,894	3,831	3,387
	Negated	2,976	1,064	913
	Non-negated	11,392	3,895	3,596
Relations	**Inter- and Intra-sentence** $\oplus$	374	128	91
	Inter- and Intra-sentence $\ominus$	159,931	56,103	52,252
	Intra-sentence $\oplus$	332	113	82
	Intra-sentence $\ominus$	12,877	5,312	3,756

Table 3.3: Quantitative description of the IxaMed-E corpus. Positive relations (denoted as $\oplus$) refer to ADRs while negative relations ($\ominus$) refer to non-ADRs.

3.2.2 Unannotated corpora

For the use of supervised machine learning approaches employed in ADR extraction, it is necessary to have the corpus annotated. However, useful features can be extracted from unannotated corpora (e.g. embeddings). To this end, we explored two different unannotated corpora written in Spanish that can be divided in in-domain corpus and out-domain corpus, described in the sections below.

In-domain corpus

The in-domain corpus, referred to as uEHR, that stands for unannotated EHRs. This also comprises EHRs from the Galdakao-Usansolo and Basurto hospitals, but they are not the same as those used in the annotated corpora. Table 3.4 shows the number of documents, word-forms and vocabulary.

uEHR	
Domain	in-domain
Documents	190,130
Word-forms	109,618,393
Vocabulary	286,984

Table 3.4: Quantitative description of uEHR, the in-domain corpus used to generate the embeddings.

Out-domain corpus

The out-domain corpus is the Spanish Billion Word Corpus and Embeddings (SBWCE) (Cardellino, 2016). This was obtained from the web and comprises journalistic texts, legislative texts, medical texts from the IULA Treebank, dumps from Wikipedia, etc. Table 3.5 shows the number of word-forms and vocabulary. Note that we are unaware of the number of documents of this corpus because this information was not provided by the authors.

SBWCE	
Domain	out-domain
Documents	-
Word-forms	1,420,665,810
Vocabulary	1,000,653

Table 3.5: Quantitative description of SBWCE, the out-domain corpus used to generate the embeddings.

3.3 Evaluation

In order to assess the quality of the ADR extraction models inferred, we turned to widely used evaluation schemes and metrics mentioned below.

3.3.1 Evaluation schemes

The evaluation can be done according to different evaluation schemes. In this work we employed **hold-out** and **k-fold cross-validation** (Manning et al.,

1999). The hold-out evaluation would enable us to expect how would perform the model on different data. The k-fold cross-validation would perform better in cases where the data is not large enough.

Specifically, during the experimentation done in Chapter 4, Chapter 5 and Chapter 6 we based on the hold-out evaluation using the train, dev and test sets described in the previous section.This evaluation was done in two ways: first training with the train set and evaluating with the dev set and next, training with the train and dev sets and evaluating with the test set. Finally, in Chapter 7 we also corroborated the results using stratified 10-fold cross-validation.

3.3.2 Evaluation metrics

The predictions given by a model can be correct or incorrect depending on the real class. These situations are represented through a confusion matrix, as shown Table 3.6. This confusion matrix corresponds to a binary classification problem, as is our case. Following widely used notation, the positive class ($\oplus$) indicates an ADR relation, the negative class ($\ominus$) indicates an unrelated pair and the values of the confusion matrix are: True Positive (TP), False Negative (FN), False Positive (FP), True Negative (TN).

PREDICTED			
$\oplus$	$\ominus$		
TP	FN	$\oplus$	REAL
FP	TN	$\ominus$	

Table 3.6: Confusion matrix that presents the number of instances predicted by the system as either positive or negative together with their real class.

With the values of the confusion matrix we calculated the evaluation metrics commonly reported (Manning et al., 1999; Dalianis, 2018): **precision** (denoted as P), **recall** (R) and **f-measure** (F). These are shown in expressions (3.1), (3.2) and (3.3) respectively.

$$P = \frac{TP}{TP + FP} \tag{3.1}$$

$$R = \frac{TP}{TP + FN} \tag{3.2}$$

$$F = 2\frac{P \cdot R}{P + R} = \frac{2TP}{2TP + FN + FP} \tag{3.3}$$

In this work, the precision, recall and f-measure were obtained for the positive and negative classes and their averages: **weighted-average** (denoted as W. Avg.), **micro-average** (Micro Avg.) and **macro-average** (Macro Avg.). The averages for precision are given in expressions (3.4), (3.5) and (3.6), where C is the number of classes and L is the number of instances, being L_i the number of instances of class i.

$$W.\ Avg.\ P = \frac{1}{L}\sum_{i=1}^{C} L_i \frac{TP_i}{TP_i + FP_i} \tag{3.4}$$

$$Micro\ Avg.\ P = \frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C}(TP_i + FP_i)} \tag{3.5}$$

$$Macro\ Avg.\ P = \frac{1}{C}\sum_{i=1}^{C} \frac{TP_i}{TP_i + FP_i} \tag{3.6}$$

The averages for recall are given in expressions (3.7), (3.8) and (3.9).

$$W.\ Avg.\ R = \frac{1}{L}\sum_{i=1}^{C} L_i \frac{TP_i}{TP_i + FN_i} \tag{3.7}$$

$$Micro\ Avg.\ R = \frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C}(TP_i + FN_i)} \tag{3.8}$$

$$Macro\ Avg.\ R = \frac{1}{C}\sum_{i=1}^{C} \frac{TP_i}{TP_i + FN_i} \tag{3.9}$$

The averages for f-measure are given in expressions (3.10), (3.11) and (3.12).

$$W.\ Avg.\ F = \frac{1}{L}\sum_{i=1}^{C} L_i \frac{2TP_i}{2TP_i + FN_i + FP_i} \tag{3.10}$$

$$Micro\ Avg.\ F = \frac{2\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C}(2TP_i + FN_i + FP_i)} \tag{3.11}$$

$$Macro\ Avg.\ F = \frac{1}{C}\sum_{i=1}^{C}\frac{2TP_i}{2TP_i + FN_i + FP_i} \tag{3.12}$$

Weighted-averaging computes the metric using the confusion matrix of each class and calculating the weighted mean where the weights are the number of instances of each class. Micro-averaging computes the metric using a single confusion matrix obtained by summing each value of the confusion matrix for all the classes. Macro-averaging computes the metric using the confusion matrix of each class and calculating the unweighted mean (Manning et al., 1999; Dessì et al., 2018). The macro-average treats all classes equally, by contrast, the weighted-average and the micro-average favor densely populated classes (Sokolova and Lapalme, 2009). Given the unequal distribution of the class of our corpora, among all the averages, we would pay more attention to the macro-averaged results. In this work we focus mainly on the results obtained for the positive class because it would give a more realistic assessment of the ADR extraction. The aforementioned metrics were calculated following the definition given by Sokolova and Lapalme (2009) for ADR detection, using Scikit-learn libraries of Python (Pedregosa et al., 2011) and corroborating the results of the classes with Weka libraries of Java (Hall et al., 2009)

Other useful measures for the tasks with skewed classes are the **ROC curve** and the **AUC** given that they are not sensible to data distributions (Fawcett, 2006). ROC graphs show how the True Positive Rate (TPR) changes against the False Positive Rate (FPR) obtained with different thresholds (Manning et al., 1999; Fawcett, 2006). The AUC was used as main evaluation metric in several related works (Karlsson et al., 2013; Zhao et al., 2014, 2015; Henriksson et al., 2015b). These metrics are also given in this work and were calculated following using Scikit-learn libraries of Python (Pedregosa et al., 2011).

3.4 Concluding remarks

Obtaining EHRs is difficult due to the fact that they are subject to strict confidentiality regulations. These are written in a free style generating a wide

lexical variability, which is reflected in the OOVs of the dev and test sets of each corpus. Annotating EHRs is complex, lots of the entities involved in the ADRs comprise more than one word and these can be situated in a long distance from one another. Furthermore, the ADRs are not frequent events, which leads to unequal distribution of the class, that is, class imbalance.

Selecting the way of evaluating the ADR extraction is not straightforward and there is not a single trend in related works. Depending of the size of the corpus, the distribution of the class or the algorithms, one of the evaluation metrics can be better than the rest. For example, the weighted-average and the micro-average favor densely populated classes, but the macro-average treats all classes equally. Then, the latter would be preferable in cases with class imbalance as ours.

4 Adverse Drug Reaction detection with symbolic representations and Random Forest

4.1 Introduction

In this work the goal is to detect ADRs. In order to decide how to start to tackle this task, we revised related works and we observed that, in the first attempts made to detect ADRs with models inferred using machine learning, the authors employed symbolic representations. These symbolic representations were discrete features (e.g. word-form, lemma, POS). Furthermore, the classifiers widely used were NB, SVM and RF. For example, Aramaki et al. (2010) employed the entities, the words and the distance between them. Miura et al. (2010) included features such as the morphemes between the entities or the order of the entities involved in the ADR. In both cases, the model was inferred with the SVM classifier. Gurulingappa et al. (2012a) employed the tokens of the sentences and their POS tags, lemmas and flags indicating if a token is a part of a named entity or not. The model was inferred using SVM. Karlsson et al. (2013) used features that correspond to the presence and temporality of different drugs and diagnoses codes, obtaining better results with the RF classifier than with the J-Rip rule learner. Patki et al. (2014), apart from n-grams, incorporated as features the synonyms of the terms and sentiment scores. Ginn et al. (2014) used a vector with the number of times that each word appeared. In both cases, the NB and SVM classifiers were used. Zhao et al. (2014, 2015) employed clinical measures and

clinical codes. RF outperformed the additional classifiers that were explored.

As a result, we decided to find a suitable symbolic representation and classifier to detect ADRs following the suggestions given in related works. To do this, we had to take into account that we wanted to detect drug-disease pairs and not only the disease or the presence of ADRs. We also had to consider that we addressed intra-sentence and inter-sentence ADRs, given that following with the annotations of the experts, the drugs and the diseases related as ADRs were either in the same sentence or in different sentences. By contrast, the majority of related works (Aramaki et al., 2010; Miura et al., 2010; Li et al., 2015) only detected intra-sentence ADRs. In our opinion, both considering the ADRs as relations and finding inter-sentence ADRs made the task more challenging. Furthermore, the experiments were done with the gold mentions (the entities manually annotated by the experts) as in (Aramaki et al., 2010; Miura et al., 2010), given that our first goal is to develop the ADR detection (as mentioned in Section 1.3).

All in all, in this chapter our aim is to address the following research questions:

Research Question 1

Which are appropriate symbolic features for ADR representation to aid machine learning algorithms?

Research Question 2

To what extent are supervised machine learning approaches appropriate for ADR detection given that ADRs are infrequent relations?

The rest of the chapter is organized as follows: Section 4.2 describes the features for the symbolic characterization. Section 4.3 explains the selected classifier. Section 4.4 explains the approaches used to tackle the class imbalance problem. Section 4.5 gives the experimental results. Section 4.6 provides the final conclusions.

4.2 Symbolic characterization

In view of the related works mentioned in the introduction (Section 4.1), we decided to explore symbolic features in our characterization. To obtain some of them we needed the morphosyntactic and semantic analyzer FreeLing-Med (Oronoz et al., 2013). As a result, the terms in SNOMED CT with their

corresponding semantic tag (substances, disorders, procedures, findings), the medical abbreviations and the brand-drug names from the drug database Bot PLUS were identified. Morphological and syntactic pieces of information were also provided, such as the lemma or the POS tag.

Our set of features is described below together with the values that these would take to represent the drug-disease pair '*esteroideo - descompensación hiperglucémica*'. Figure 4.1 shows graphically this example, which was extracted from the sentence '*A consecuencia del tratamiento esteroideo se produce descompensación hiperglucémica que precisa tratamiento con insulinización*'.

1. **Entity-words and context-words**: The medical-entity terms for both drug and disease together with the left and right context-words were kept. The context was a window of size k, with k=3 in our case, an intermediate value among the sizes used in other works (Gurulingappa et al., 2011; Li et al., 2013). Overall, this yielded 14 features. Moreover, we also experimented with different values of k in Section 4.5. In our example, the values of these features are given in the first line of Figure 4.1 ("word-form"). The entity-words are *descompensacion-hiperglucemica* and *esteroideo* and the context-words are *esteroideo*, *se*, *produce*, *que*, *precisa*, *tratamiento*, *consecuencia*, *del*, *tratamiento*, *se*, *produce*, *descompensacion*.

2. **Entity-lemmas and context-lemmas**: These features corresponded to the lemmas of the aforementioned entity-words and context-words (14 nominal features). In our example, the values of these features are given in the second line of Figure 4.1 ("lemma"). The entity-lemmas are *'descompensacion - hiperglucemica'*, *esteroideo* and the context-lemmas are *esteroideo*, *se*, *producir*, *que*, *precisa*, *tratamiento*, *consecuencia*, *del*, *tratamiento*, *se*, *producir*, *descompensacion*.

3. **Entity-POS and context-POS**: These were 14 features that corresponded to the POS of the aforementioned entity-words and context-words. In our example, the values of these features are given in the third line of Figure 4.1 ("POS"). The entity-POS tags are 'NCFS000 - NCFS000', NCMS000 and the context-POS are NCMS000, P00CN000, VMIP3SO, PR0CN000, VMIP3SO, NCMS000, NCMS000, 'SPS00 - DA0MS0', NCMS000, P00CN000, VMIP3SO, NCFS000.

4. **Drug family**: A nominal feature that corresponded to the ATC classification (World Health Organization, 2003) of the drug entity. In our example, the value of this feature is given in the fourth line of Figure 4.1 ("drug family'). In this case, the drug is not found in the list of drug families and the value is *'NO HAY FAMILIA'*.

5. **Presence/absence of other drugs**: A numeric feature to indicate whether there were other drugs in the context of the target drug and disease. Specifically, the value of this feature was the number of drugs in the context. In our example, the value of this feature is given in the fifth line of Figure 4.1 ("drugs"). In this case, there is one drug in the left context of the disease, which is the drug of the ADR, and the value is 1.

6. **Negation modifiers**: These were two binary features to determine whether each entity of the pair was negated. If we do not have these annotations, it is possible to use a system to automatically detect the negated entities. Specifically, we explored the detection of negated entities in EHRs with two approaches: 1) the rule-based system NegEx (Santiso et al., 2017) and 2) a CRF classifier (Santiso et al., 2018b) (turn to Appendix A to see more information about the detection of negated entities). According to the example, the values of these features are given in the sixth line of Figure 4.1 ("negation"). Given that the entities are not negated, the values are *noNegado* for the disease and *noNegado* for the drug.

7. **Trigger words**: A binary feature that indicated the presence or absence of trigger words like "*causado por*" (caused by), "*relacionado con*" (related with), "*secundario a*" (secundary to), "*debido a*" (due to) between the drug and the disease of the pair. In our example, the value of this feature is given in the seventh line of Figure 4.1 ("trigger-word"). In this case, there are not trigger-words between the entities and the value is false.

8. **Distances**: The number of characters and sentences from the drug entity to the disease entity (2 numeric features). The distance could be negative if the drug entity preceded the disease entity. The intuition was that if a drug and a disease entities were close, the probability of forming an ADR was higher. In our example, the values of these

features are given in the eighth line of Figure 4.1 ("distances"). The distance given in characters is -12 and the distance given in sentences is 0. Note that the distance in sentences is meaningless if we restrict to detect intra-sentence ADRs.

	A	consecuencia	del	tratamiento	esteroideo	se	produce
word-form		consecuencia	del	tratamiento	esteroideo	se	produce
lemma		consecuencia	de-el	tratamiento	esteroideo	se	producir
POS		NCMS000	SPS00-DA0MS0	NCMS000	NCMS000	P00CN000	VMIP3S0
drug family					NO HAY FAMILIA		
drugs						1	
negation					noNegado		
trigger-word						false	
distances						-12 | 0	

...

	descompensación	hiperglucémica	que	precisa	tratamiento	con	insulinización
word-form	descompensacion-hiperglucemica		que	precisa	tratamiento		
lemma	descompensacion-hiperglucemica		que	precisar	tratamiento		
POS	NCFS000-NCFS000		PR0CN000	VMIP3S0	NCMS000		
drug family							
drugs							
negation	noNegado						
trigger-word							
distances							

Figure 4.1: Scheme of the features used for the symbolic characterization of the ADR '*esteroideo - descompensación hiperglucémica*' present in the sentence 'As a result of the steroidal treatment, it was produced an hyperglycaemic decompensation that needs treatment with insulinization'. The features related with the entities are highlighted in dark blue and the features related with the context in light blue.

We applied **attribute selection** to select the most relevant attributes using Information Gain (Quinlan, 1986) with respect to the class, that is, the higher the Information Gain the better the correlation with the class. We selected the 20 most relevant ones because, inspecting the relevance of the features taking into account all the instances, we observed that around this position the value of the Information Gain was approximately a half of the one obtained by the best feature (see Table 4.1). In this way, we also took into account the word-form of the drug.

Ranking	InfoGain	Feature
1	0.013641	distance (characters)
2	0.012797	distance (sentences)
3	0.012522	disease
4	0.011723	disease lemma
5	0.010339	2nd word of the context after the disease
6	0.010091	2nd lemma of the context after the disease
7	0.009826	2nd word of the context before the disease
8	0.009760	3rd word of the context after the disease
9	0.009628	2nd lemma of the context before the disease
10	0.009490	3rd lemma of the context after the disease
11	0.009291	1st word of the context before the disease
12	0.008945	1st lemma of the context before the disease
13	0.008798	3rd word of the context before the disease
14	0.008641	3rd lemma of the context before the disease
15	0.008441	1st word of the context after the disease
16	0.008253	1st lemma of the context after the disease
17	0.007695	3rd word of the context after the drug
18	0.007457	3rd lemma of the context after the drug
19	0.007359	drug
20	0.007275	2nd word of the context before the drug
21	0.007196	3rd word of the context before the drug
22	0.007066	2nd lemma of the context before the drug
23	0.007005	drug lemma
24	0.006986	2nd word of the context after the drug
25	0.006936	2nd lemma of the context before the drug
26	0.006827	3rd lemma of the context before the drug
27	0.006048	1st word of the context after the drug
28	0.005957	1st lemma of the context after the drug
29	0.005491	disease POS
30	0.004794	1st word of the context before the drug
31	0.004575	1st lemma of the context before the drug
32	0.003876	2nd POS of the context before the drug
33	0.003862	1st POS of the context before the disease
34	0.003538	3rd POS of the context before the drug
35	0.003493	1st POS of the context before the drug

(Continued on next page)

Ranking	InfoGain	Feature
36	0.003453	2nd POS of the context after the drug
37	0.003273	3rd POS of the context after the drug)
38	0.003225	1st POS of the context after the drug
39	0.002778	2nd POS of the context after the disease
40	0.002690	1st POS of the context after the disease
41	0.002558	drug POS
42	0.002556	3rd POS of the context before the disease
43	0.002270	3rd POS of the context after the disease
44	0.001800	drug family
45	0.001737	2nd POS of the context before the disease
46	0.001378	negated entity
47	0.000315	drugs
48	0.000188	trigger-words
49	0.000000	negated drug

Table 4.1: Ranking of the features according to the Information Gain (denoted as "InfoGain"). These features are those created for the symbolic representation of the intra-sentence as well as inter-sentence ADR candidates in the IxaMed-GS corpus.

Furthermore, the feature values allowed OOV words, in a way that it is possible to include in the evaluation set instances with values that do not appear in the features of the train set.

4.3 The choice of classifier

With the aim of checking if our set of features was useful to detect the ADRs in our EHRs, we developed an experiment using the RF classifier (Breiman, 2001). In the introduction (see Section 4.1) we already commented that this classifier was used in related works about ADR detection. Apart from this, the selection of this classifier was motivated by preliminary experiments that we developed in the degree thesis and in the master thesis, where we observed that RF was able to carry out the ADR detection (Santiso et al., 2014) and resulted more robust for this task in comparison with other classifiers such as

SVM (Casillas et al., 2016b). Note that, since this was explained in previous works, we do not go into detail and we summarize the lessons learned.

RF combines a number of decision trees being each tree built on the basis of the C4.5 algorithm and with a sample of data obtained using bagging, there are other variants that are built based on bootstrapping. RF has the characteristic of introducing some randomness to split the nodes of each tree. Particularly, each time a node is generated in the tree, instead of choosing the attribute that maximizes the Information Gain, it selects the best attribute among a random subset of features. The bagging and the random features selection help to avoid overfitting and, for this reason, it obtains good generalization ability (Qi, 2012). Figure 4.2 shows the general architecture of the RF algorithm. X is the instance, n is the number of trees, c_1, c_2 and c_n are the class assigned to the instance for each tree and c is the final class assigned to the instance selected by voting.

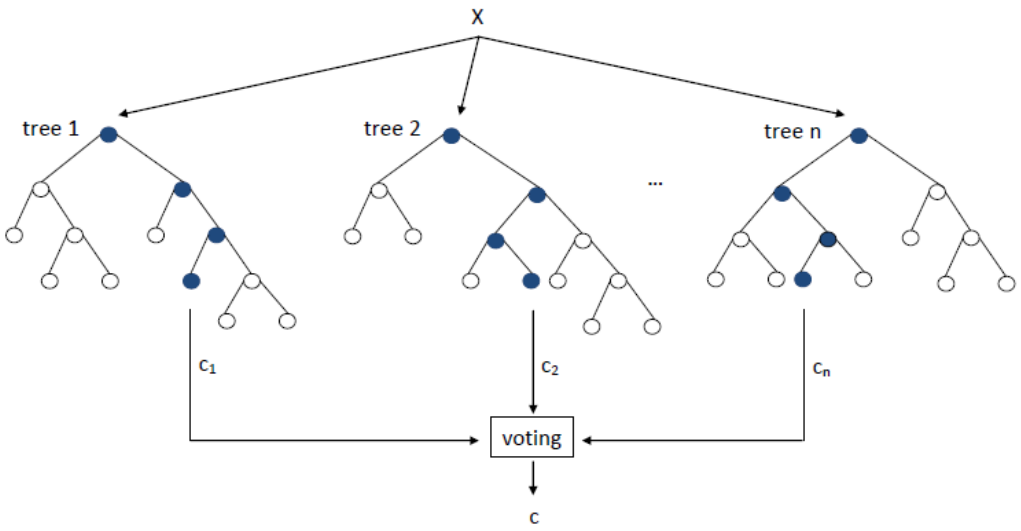


Figure 4.2: Scheme of the general architecture of the Random Forest algorithm.

In the experiment developed for this first approximation, all the instances were predicted as non-ADRs. That is to say, the classifier assigned the negative class to all of them. Inspecting the instances of the set used to infer the model we observed the presence of skewed class distribution and our impression was that the ADRs had not been detected due to the class im-

balance problem. This is the highly unequal distribution of instances of each class, there is a big amount of examples of one class (normal cases) and a small amount of the other one (anomalous cases) (Chandola et al., 2009). Naturally, this imbalance was higher taking into account intra-sentence and inter-sentence ADR candidates than taking into account just intra-sentence ADR candidates. In Figure 4.3 we can see that there are 16 inter-sentence ADRs (the entities of the pair are in different sentences) and 53 intra-sentence (the entities of the pair are in the same sentence).

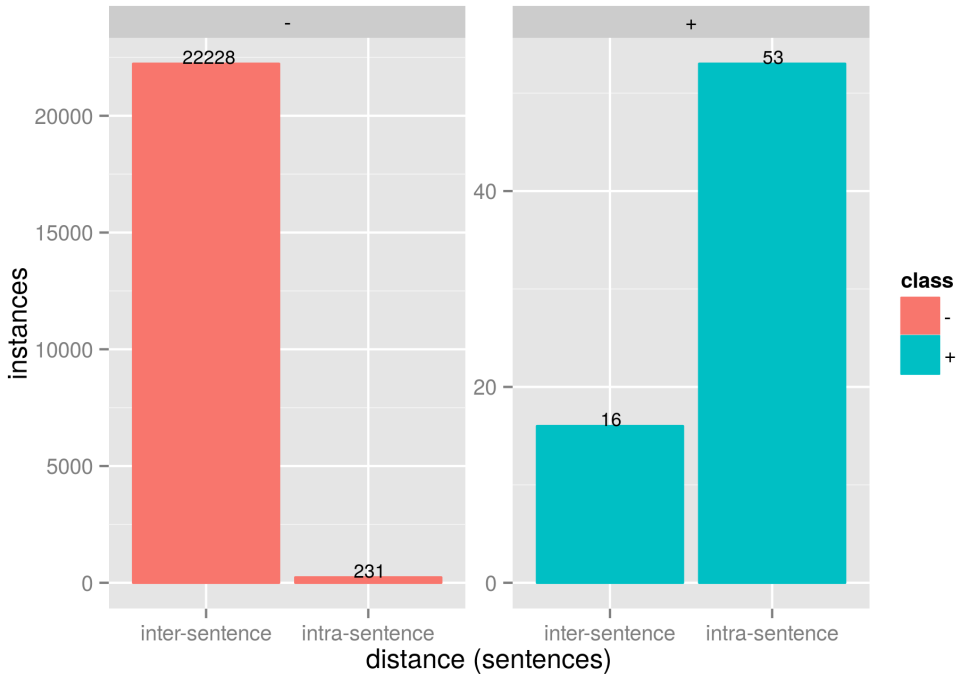


Figure 4.3: Histogram of the number of inter-sentence and intra-sentence instances in the train set of the IxaMed-GS corpus for the positive class (denoted as $\oplus$) and the negative class ($\ominus$). Note that the number of instances is represented with a different scale for each class.

According to this information, we thought that it was necessary to tackle the class imbalance using the approaches described below (see Section 4.4).

4.4 Techniques to overcome the class imbalance

In the last years, several attempts were made to overcome the class imbalance (He and Garcia, 2009; Nanni et al., 2015). There are related works devoted to find ADRs with symbolic representations that had also to tackle this problem. For example, Aramaki et al. (2010) and Miura et al. (2010) observed that the few positive data, worsened the performance of their model. Sohn et al. (2011) employed up-sampling to increase the number of positive instances (imbalance ratio of 1:7). Botsis et al. (2011) and Patki et al. (2014) applied cost-sensitive learning (imbalance ratio of 1:25 and 1:3 respectively). Ginn et al. (2014) divided the corpus in three datasets with different distributions of the class (imbalance ratio ranging from 1:1 to 1:2). Zhao et al. (2014) and Zhao et al. (2015) decided to use AUC as evaluation metric since it is robust against distribution changes (higher imbalance ratio of 1:43). Sarker and Gonzalez (2015) used different corpora to extract ADRs and observed that the combination of the different datasets helps to improve the performance in presence of class imbalance. In our work, we turned to different techniques that could be divided in the following groups:

- **Sampling**: The sampling methods add or remove instances, which allows to obtain an equal distribution of the class and avoid the bias (Estabrooks et al., 2004). These methods are applied only on the training set. On the one hand, in order to make uniform the class distribution, we resorted to re-sample (Hall et al., 2009) and spread-subsample (Hall et al., 2009). On the other hand, we employed three self-implemented techniques to reduce the number of instances of the majority class. These techniques do not uniform the class distribution necessarily, that is to say, we can obtain an imbalanced distribution again, although with a lower ratio. They were the following: numeric sub-sample (discards those instances where the distances in term of words between the entities are higher or lower than a given threshold), nominal sub-sample (discards those instances that correspond to the same drug and disease within the same document) and negation sub-sample (discards those instances that contain a negated entity).

- **Cost-sensitive learning**: Cost-sensitive learning (Domingos, 1999) assigns stronger penalties to instances in which the majority class is predicted incorrectly in each iteration of the inference stage. We made the classifier cost-sensitive by applying costs to each type of error as (Elkan,

2001) proposed. The weights that we assigned to the False Positives (FPs) were the proportion of positive instances (the number of positive instances into the total number of instances) and to the False Negatives (FNs) the proportion of negative instances (the number of negative instances into the total number of instances).

- **Ensemble learning**: Ensemble learning combines different learning approaches in an attempt to improve the definition of the decision boundary for each region, paying attention to the minority class (Galar et al., 2012). On the one hand, we used ensemble techniques that generate multiple versions of a classifier, such as bagging (Breiman, 1996) and boosting (Freund and Schapire, 1996). On the other hand, we used ensemble techniques that combined the outputs obtained by different classifiers, such as stacking (Wolpert, 1992). Furthermore, we employed other algorithms to combine the predictions of different classifiers, such as weighted voting, majority voting and OR voting.

- **One-class classification**: The one-class classification (Chang and Lin, 2011) treats the instances of the minority class as outliers, inferring regularities about the majority class and focusing on discarding the minority class (Lee and Cho, 2006). For our task the outliers would be the ADRs, given that this is the minority class.

These techniques to overcome the class imbalance were applied in different ways: i) using the individual techniques, ii) using combinations of the individual techniques, and iii) using ensembles of the individual techniques and their combinations. As a result, the approaches explored in Table 4.2 were explored.

		Sampling						Cost-sensitive	Ensemble		One-class	Approach
		re-sample	spread sub-sample	numeric sub-sample	nominal sub-sample	negation sub-sample			bagging	boosting		
Individual		✓										1
			✓									2
				✓								3
					✓							4
						✓						5
								✓				6
									✓			7
										✓		8
											✓	9
Combination		✓						✓				10
		✓						✓	✓			11
				✓	✓							12
		✓		✓	✓							13
		✓				✓						14
		✓				✓		✓				15
Ensemble	weighted voting	✓										16
		✓						✓				
		✓						✓	✓			
											✓	
	majority voting	✓										17
		✓						✓				
		✓						✓	✓			
											✓	
	OR voting	✓										18
		✓						✓				
		✓						✓	✓			
											✓	
	stacking	✓										19
		✓						✓				
		✓						✓	✓			
											✓	

Table 4.2: Different techniques to overcome the class imbalance (applied individually, in combination or with ensembles) produced different experimental approaches. The rows of each ensemble method correspond to the employed approaches ("Individual" or “Combination”).

4.5 Results

In this section we show the results obtained with the different approaches proposed to tackle the class imbalance. Before this, we consider important to bear in mind that the models were inferred with RF (Breiman, 2001) as main classifier. This was implemented in Java with the Weka libraries (Hall et al., 2009). To improve the performance of the classifier, we carried out a parameter selection to choose the number of trees to be generated and the number of attributes to use in the randomization process. The fine-tuning was carried out exploring 10 values close to the values by default. In addition, we removed the accents and we transformed all the words to lowercase. To assess the models we used the IxaMed-GS corpus and the hold-out evaluation scheme (see Section 3.3.1).

Figure 4.4 shows the f-measure of the positive class for the dev set obtained for the detection of **inter-sentence and intra-sentence** ADRs tackling the class imbalance. From our experimental framework, we observed that there were two approaches that obtained better results than the rest: i) re-sample (approach 1) and ii) combination of re-sample and cost-sensitive (approach 10). The f-measure for the positive class was 11.2 and 11.0 respectively. Additionally, we observed that with the approach 1 we obtained 178 FPs and 32 FNs and with the approach 10 we obtained 208 FPs and 30 FNs. Given the difference in the number of FPs (30 FPs), we found that the application of re-sample before using the cost-sensitive classifier resulted robust to overcome this highly skewed classification task.

The application of the approaches to overcome the class imbalance was of much help. In fact, the experiments developed without tackling the imbalance (the baselines) performed worse than with the application of the approach that resulted the most effective to overcome the class imbalance. This happened for both inter- and intra-sentence relations and for only intra-sentence relations. Table 4.3 provides detailed results of these experiments. The best approach using **inter- and intra-sentence** scope (approach 10) entails re-sample and cost-sensitive and its corresponding baseline is **Baseline-0**. The best approach using **intra-sentence** scope (approach 1) entails re-sample and its corresponding baseline is **Baseline-1**. It is obvious that the results obtained at sentence level outperformed those obtained in the experiments that take into account all the relations. Specifically, the f-measure of the positive class varied from 11.0 to 46.2. In addition, for intra-sentence relation, the use of re-sample outperformed the baseline from 37.8 to 46.2.

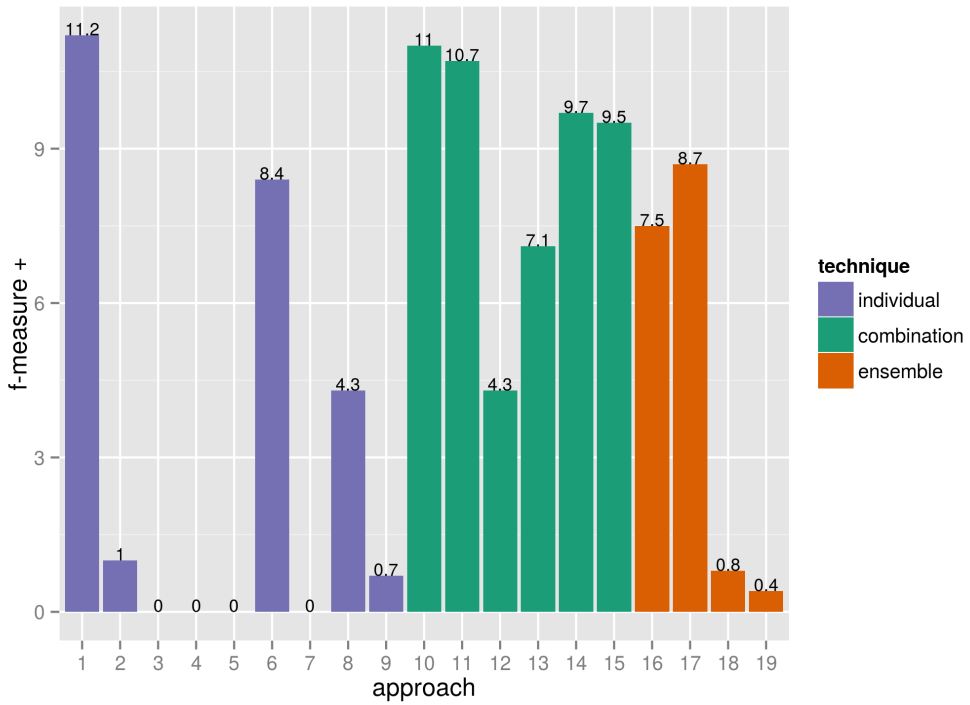


Figure 4.4: F-measure for the positive class obtained in the experiments developed with the approaches of Table 4.2 to overcome the class imbalance. The models were inferred with the train set and evaluated with the dev set of IxaMed-GS corpus using the Random Forest classifier and exploring inter- and intra-sentence relations.

Scope	Approach	Precision	Recall	F-measure	Class
Inter- and Intra-sentence	Baseline-0	0.0	0.0	0.0	⊕
		99.7	1.0	99.9	⊖
		99.5	99.7	99.6	W. Avg.
		99.7	99.7	99.7	Micro Avg.
		49.9	50.0	49.9	Macro Avg.
	Approach10	6.8	28.9	11.0	⊕
		99.8	99.0	99.4	⊖
		99.6	99.8	99.2	W. Avg.
		99.8	99.8	99.8	Micro Avg.
		53.3	63.9	55.2	Macro Avg.
Intra-sentence	Baseline-1	1.0	23.3	37.8	⊕
		85.4	1.0	92.1	⊖
		88.0	86.0	82.2	W. Avg.
		86.0	86.0	86.0	Micro Avg.
		92.7	61.7	65.0	Macro Avg.
	Approach1	54.5	40.0	46.2	⊕
		87.3	92.5	89.9	⊖
		81.3	82.9	81.9	W. Avg.
		82.9	82.9	82.9	Micro Avg.
		70.9	66.3	68.0	Macro Avg.

Table 4.3: Results of the best performing models (approaches 1 an 10 in Table 4.2) and the baselines, when are used the inter- and intra-sentence relations or just the intra-sentence relations. The models were inferred with the train set and evaluated with the dev set of the IxaMed-GS corpus using the Random Forest classifier.

In all the experiments developed before, we applied **feature selection** to select the 20 most relevant features. Surprisingly, in Table 4.4 we can observe that the results of the best performing experiment increased from 44.8 to 46.2 without the feature selection, that is, using all the features. This corroborated that the feature selection was beneficial in this case. For this experiment developed only with the intra-sentence relations, the rank of the selected features correspond to the features 3, 4, 5, 7, 6, 17, 9, 11, 19, 12, 24, 8, 23, 25, 10, 18, 13, 14, 21 and 20 in Table 4.1. These correspond to the word-forms and lemmas of the entities and their contexts.

Feature selection	Precision	Recall	F-measure	Class
without	40.5	50.0	44.8	⊕
	88.2	83.6	85.8	⊖
	79.5	77.4	78.3	W. Avg.
	77.4	77.4	77.4	Micro Avg.
	64.4	66.8	65.3	Macro Avg.
with	54.5	40.0	46.2	⊕
	87.3	92.5	89.9	⊖
	81.3	82.9	81.9	W. Avg.
	82.9	82.9	82.9	Micro Avg.
	70.9	66.3	68.0	Macro Avg.

Table 4.4: Results of the best performing model (intra-sentence ADRs and re-sample) with and without feature selection. The model was inferred with the train set and evaluated with the dev set of the IxaMed-GS corpus using the Random Forest classifier.

Given that the performance was assessed with a context-window of length 3 (see context-word features in Section 4.2), we explored the impact associated with the length of the context-window (k). These results are shown in Figure 4.5, where none of the new context-window length (2,4,6) outperformed the results obtained with length 3.

Finally, Table 4.5 gives full details of the results achieved with the best performing configuration (approach 10 in Table 4.2) for the dev and test sets. For the dev set, the f-measure of the positive class was, 46.2. For the test set, the f-measure of the positive class was 43.2.

In addition, we analysed the ROC curve and the AUC. Figure 4.6 shows the ROC curve and the AUC of the aforementioned experiments for the dev and test sets. Unexpectedly, the AUC of the test set is better than the AUC of the dev set, contrary to the happened with the f-measure.

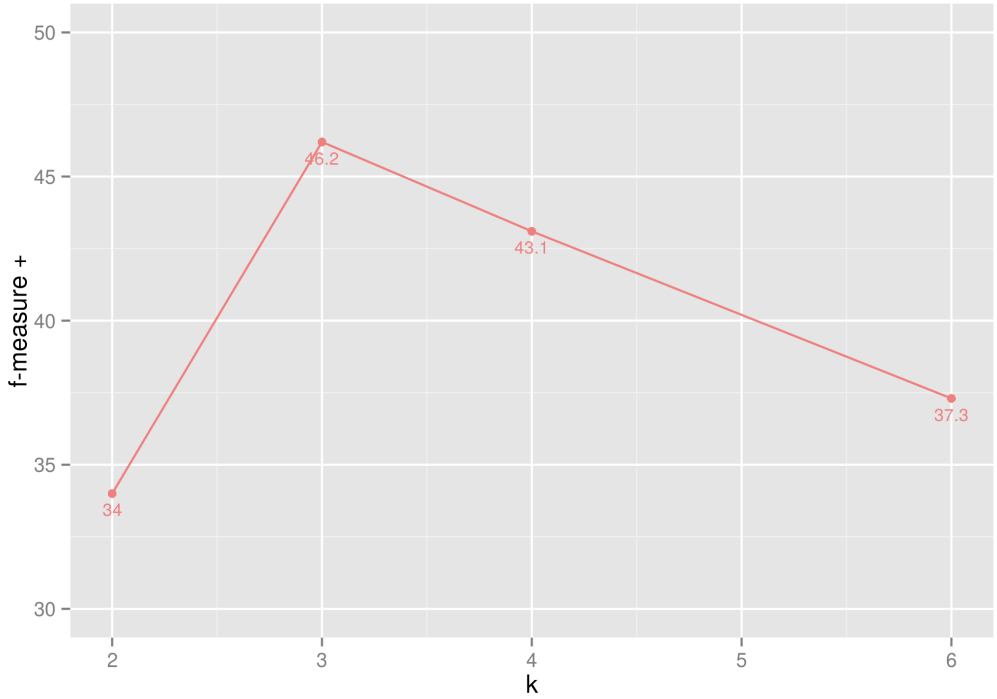


Figure 4.5: F-measure of the positive class varying the length of the context-window (k) for the best performing model (intra-sentence ADRs and re-sample). The model was inferred with the train set and evaluated with the dev set of the IxaMed-GS corpus using the Random Forest classifier. Note that the f-measure of the positive class is represented from 30.

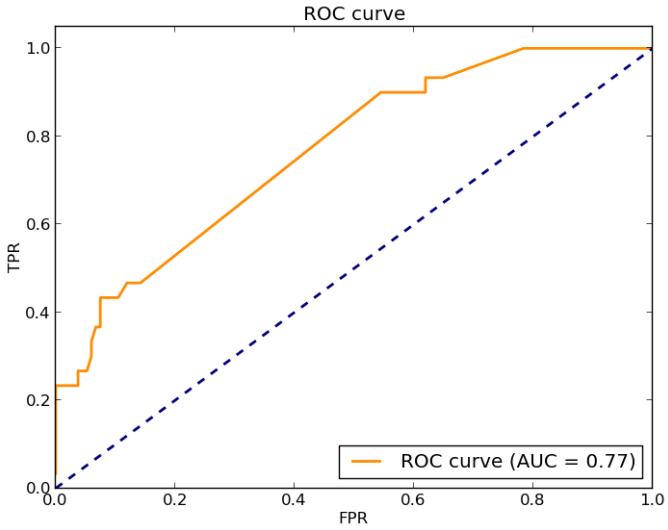
Precision	Recall	F-measure	Class
54.5	40.0	46.2	$\oplus$
87.3	92.5	89.9	$\ominus$
81.3	82.9	81.9	W. Avg.
82.9	82.9	82.9	Micro Avg.
70.9	66.3	68.0	Macro Avg.

(a) Model inferred with the train set and evaluated with the dev set.

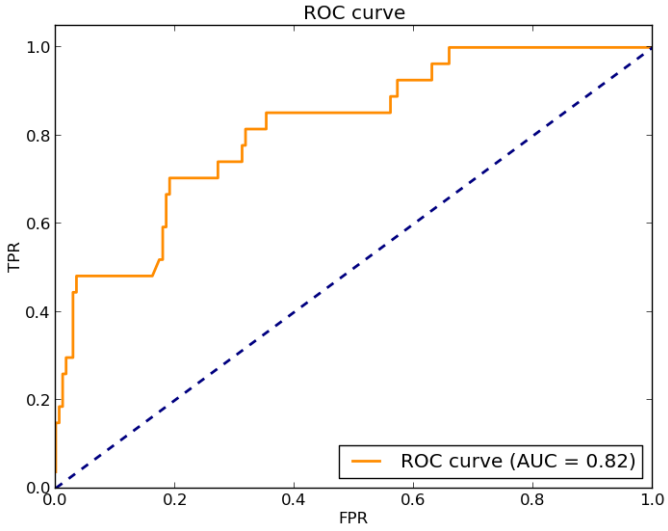
Precision	Recall	F-measure	Class
34.0	59.3	43.2	$\oplus$
92.8	82.1	87.1	$\ominus$
84.9	79.0	81.2	W. Avg.
79.0	79.0	79.0	Micro Avg.
63.4	70.7	65.2	Macro Avg.

(b) Model inferred with the train and dev sets and evaluated with the test set.

Table 4.5: Results of the best performing model (intra-sentence ADRs and re-sample) inferred with the IxaMed-GS corpus and the Random Forest classifier.



(a) Model inferred with the train set and evaluated with the dev set.



(b) Model inferred with the train and dev sets and evaluated with the test set.

Figure 4.6: ROC curve and AUC of the best experiment (intra-sentence ADRs and re-sample). The model was inferred with the IxaMed-GS corpus and the Random Forest classifier.

4.5.1 Discussion

The ADR detection was developed with the RF classifier. This classifier was already employed by other authors (Zhao et al., 2014, 2015) for ADR extraction with symbolic characterizations with promising results.

RF selects the features that it considers more relevant to generate the trees. However, applying **feature selection** strategies in advance resulted beneficial. It is possible that this helped the classifier to pay less attention to the redundant features. Among all the hand-crafted features used in our symbolic representation, the 20 most relevant features for the intra-sentence scope were the word-forms and lemmas of the entities and their contexts. By contrast, the distances are the most relevant ones when inter- and intra-sentence scope is considered, as shows Table 4.1.

Regarding the **class imbalance**, trying to classify drug-disease pairs into ADR or non-ADR relations was not straightforward due to the imbalance ratio. This was 1:222, when we took into account all the ADR candidates (inter-sentence and intra-sentence ADRs). We incorporated individual approaches to overcome the class imbalance and we learned that, in this task, individual techniques were not of much help, except for Sampling. Experiments disclosed that it was worth combining Sampling with Cost-sensitive learning. Particularly, the best results, in terms of f-measure of the positive class, were achieved with the application of re-sample before using the cost-sensitive classifier.

Restricting the ADR detection to sentence level alleviated drastically the class imbalance problem, obtaining an imbalance ratio of 1:4. Indeed, there are related works (Aramaki et al., 2010; Miura et al., 2010; Li et al., 2015) that only explored drug-disease pairs placed in the same sentence. Although some of the ADR instances were not taken into account because their entities were situated in different sentences, the imbalance reduction helped to improve the ADR detection. This improvement was notable, above all, in the precision (changing from 7.2 to 34.0 in the dev set). The use of a Sampling technique such as re-sample was also useful and made possible to capture a higher number of ADRs. Furthermore, we proved that a context-window of length 3 was optimal being the f-measure of the positive class 43.2.

We regard to the **OOVs**, these were analyzed in the intra-sentence relations. To be precise, we analyzed those values of each features in the evaluation set that do not appear among the values of each features in the training set. We observed that 26% (851 of 3,280) of the values of the dev

set were not found in the train set. This can affect negatively to the performance of the system. For example, the performance of the ADR detection could worsen using all the features because the number of values not found is higher than applying attribute selection.

So far, we distinguished between inter- and intra-sentence relations. However, in an EHR the same drug-disease pair can appear in different positions of the document. Therefore, the experts requested, in the framework of the DETEAMI and PROSAMED projects mentioned in Section 1.2, not to mark just each pair in its corresponding position in the document, but also provide them as a summary. Hence, we developed a variant of our previous system in which if a drug-disease pair appeared several times in a document and was predicted as ADR at least once, this pair is labeled as ADR in the summary (Santiso et al., 2016). The experts validated the ADR extraction with an on-line prototype of the system (Casillas et al., 2016a) and they considered that although it was useful to get a summary of each EHR, it was more precise to get the prediction for each drug-disease individually throughout the document.

4.5.2 Error analysis

After analyzing the results, we inspected the predictions made by this model. We observed that it was able to detect ADRs that were not found by the model inferred without applying re-sample. To illustrate this, see Figure 4.7, where the black arrows "Causada_por" correspond to the ADRs annotated by the experts and the red arrows "Causada_por_system" correspond to the predictions made by the system. For example, the ADRs '*hipoglucemia - septrin*' and '*hipoglucemia - timetropin*' were detected (the pairs '*hiperglucemia - novonorm*', '*disminución del apetito - trimetropin*', '*disminución del apetito - novonorm*' were incorrectly detected as ADRs). Note that the black arrows "Causada_por" correspond to the ADRs annotated by the experts and the red arrows "Causada_por_system" correspond to the predictions made by the system.

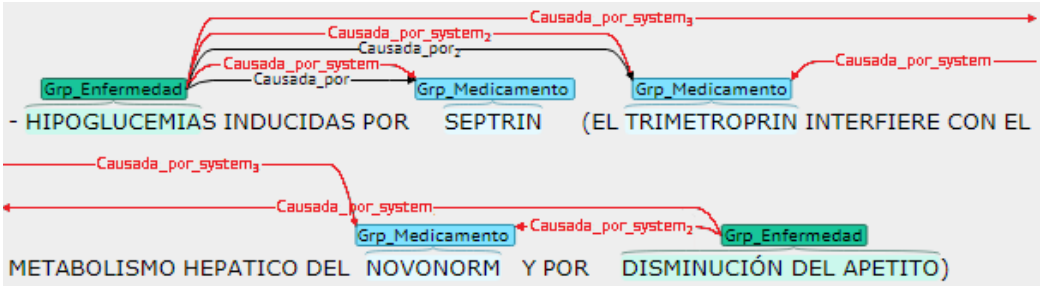


Figure 4.7: Example of sentence in which the model inferred with the symbolic characterization detected the ADRs annotated by the experts. The sentence means 'Hypoglycemias induced by septrin (the trimethoprim interferes with the hepatic metabolism of the novonorm and by decreased appetite)'.

We also observed several sources of errors. First, in very **long sentences** (around 60 words) with a wide combination of potential events (around 14), it was not infrequent to find FPs. For example, in the sentence shown in Figure 4.8 the pairs '*dm tipo 2* - *insulina lantus*', '*dislipemia* - *insulina lantus*' and '*resistencia insulina* - *insulina lantus*' were incorrectly detected as ADRs, producing 3 FPs (the ADR '*intolerancia* - *diamben*' was detected correctly).

Second, the system detected as ADR some drug-disease pairs that are related as **treatment**, yielding FPs. This could happen because sometimes the word "treatment" also appears when an adverse reaction is indicated. For example, in the sentence shown in Figure 4.9 the pairs '*reacción alérgica* - *amoxicilina-clavulanico*', '*reacción alérgica* - *esteroides*' and '*reacción alérgica* - *antibiótico*' were incorrectly detected as ADRs, producing 3 FPs. By contrast, the ADR '*reacción alérgica* - *levofloxacino*' was detected correctly.

Furthermore, in speculative sentences with medical **uncertainty** (Velupillai and Kvist, 2012), that is, sentences where the diagnosis of the doctor about an ADR is uncertain, we found FNs. For example, in the sentence given in Figure 4.10 the uncertain ADRs '*descompensación cardiaca izquierda* - *AINEs*' and '*liger empeoramiento de su función renal* - *AINEs*' were labeled by the experts, but their were not detected by the system. Our impression is that this type of error happened because the system generalizes as non-ADR some uncertain ADRs that were labeled by the experts.

In brief, the application of re-sample helped to increase the number of detected ADRs. We found as sources of FPs long sentences with a wide combination of drug-disease pairs and sentences with drug-disease pairs that are related as treatment. The sources of FNs were speculative sentences.

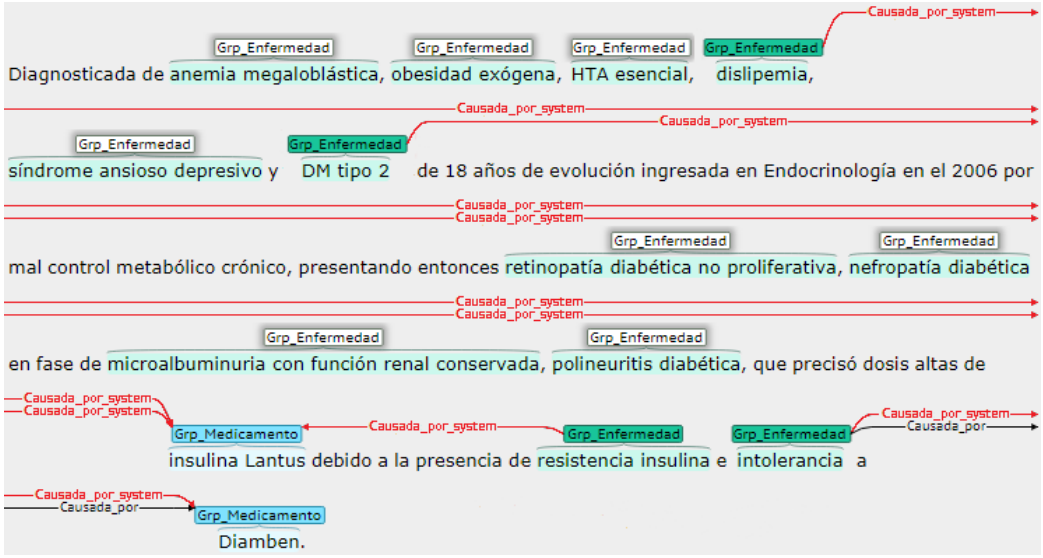


Figure 4.8: Example of long sentence in which the best performing model committed 3 FPs. The sentence means 'Diagnosed with megaloblastic anemia, exogenous obesity, essential HTN, dyslipidemia, anxiety-depressive syndrome and type 2 DM of 18 years of evolution admitted to Endocrinology in 2006 because of bad chronic metabolic control, then presenting non-proliferative diabetic retinopathy, diabetic nephropathy in microalbuminuria with preserved renal function phase, diabetic polyneuritis, which required high doses of Insulin Lantus due to the presence of insulin resistance and intolerance to Diamben.'.

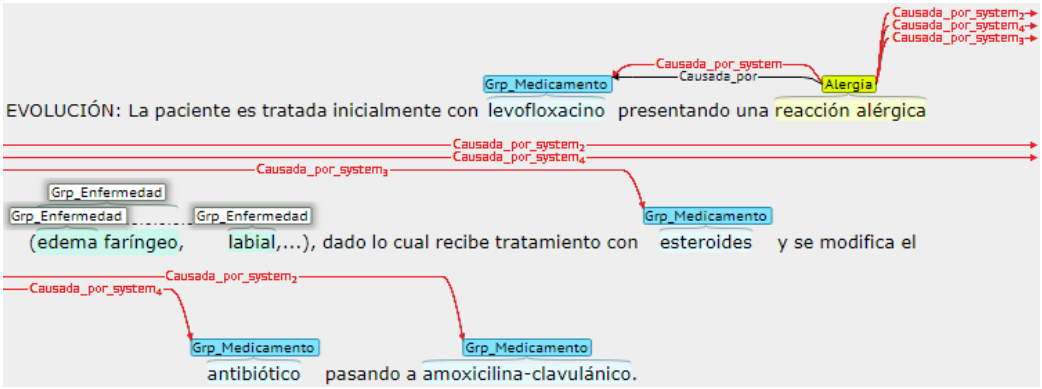


Figure 4.9: Example of sentence related with a treatment in which the best performing model committed 3 FPs. The sentence means 'Evolution: The patient is initially treated with levofloxacin showing an allergic reaction (pharyngeal, lip, ... edema), for this reason she received treatment with steroids and the antibiotic is modified changing into amoxicillin-clavulanic.'.

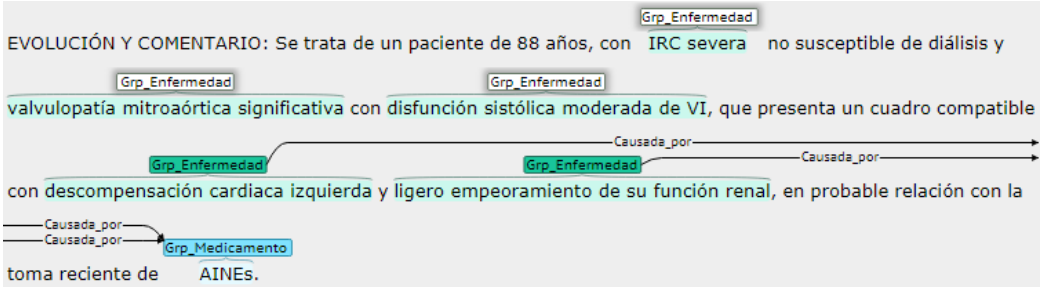


Figure 4.10: Example of speculative sentence in which the best performing model committed 2 FNs. The sentence means 'Evolution and comment: He is a patient of 88 years old, with severe CKD that is not suitable for dialysis and significant micro-aortic valvular heart disease with moderate systolic dysfunction of LV, that presents symptoms compatible with left heart failure and slight deterioration of his kidney function, in probable relation with the recent take of NSAIDs.'.

4.6 Conclusions

4.6.1 Concluding remarks

In this chapter we developed an ADR detection system using a symbolic characterization and the RF classifier. We started exploring inter-sentence as well as intra-sentence ADRs, obtaining extremely poor results due to the imbalanced datasets. We explored different approaches to overcome the class imbalance and we obtained some improvement of the performance with the combination of re-sample and cost-sensitive learning. After that, we restricted the ADR detection to intra-sentence relations. In other words, we only used drug-disease pairs placed in the same sentence. This yielded a better performance of the ADR detection system. Thereafter, we decided to restrict our system to the sentence-level ADR detection.

According to this experimentation, we answered to the following research questions:

Research Question 1

Which are appropriate symbolic features for ADR representation to aid machine learning algorithms?

The symbolic characterizations together with the machine learning algorithms can be used to detect ADRs in EHRs written in Spanish. When the ADR extraction is developed as a relation extraction task, the causative drug and the caused disease are involved in the relation. If inter-sentence and intra-sentence ADRs are taken into account, features related with the distances between the entities involved result relevant for the task. If the ADR detection is focused on intra-sentence ADRs, the word-forms and the lemmas of the entities and their contexts are more relevant.

Research Question 2

To what extent are supervised machine learning approaches appropriate for ADR detection given that ADRs are infrequent relations?

With imbalanced datasets the machine learning algorithms such as Random Forest tend to be biased and learning to predict the minority class is complex. The application of approaches to overcome the class imbalance improves the performance of the ADR detection model to

find inter- and intra-sentence ADRs. However, inter- and intra-sentence ADRs is ambitious and the restriction to intra-sentence ADRs improves drastically the detection of ADRs.

Open question. The symbolic characterization employed in this case can be problematic to obtain the lemmas or the POS used as features since the EHRs contain misspellings or abbreviations that can make their processing difficult. In addition, the generalization over unseen words is not robust with symbolic features. Therefore, in the next chapter we explore dense characterizations based on word-embeddings to represent the drug-disease pairs.

4.6.2 Publications

This work lead to the following publications:

1. Sara Santiso, Arantza Casillas, Alicia Pérez, Maite Oronoz, and Koldo Gojenola. Adverse drug event prediction combining shallow analysis and machine learning. In *Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis (Louhi)*, pages 85–89, Gothenburg, Sweden, April 26-30 2014.

2. Arantza Casillas, Alicia Pérez, Maite Oronoz, Koldo Gojenola, and Sara Santiso. Learning to extract adverse drug reaction events from electronic health records in Spanish. *Expert Systems with Applications*, 61:235–245, 2016.

3. Sara Santiso, Arantza Casillas, Alicia Pérez, Maite Oronoz, and Koldo Gojenola. Document-level adverse drug reaction event extraction on electronic health records in Spanish. *Procesamiento del Lenguaje Natural*, 56:49–56, 2016.

4. Arantza Casillas, Arantza Díaz de Ilarraza, Kike Fernandez, Koldo Gojenola, Maite Oronoz, Alicia Pérez, and Sara Santiso. IXAmed-IE: on-line medical entity identification and ADR event extraction in Spanish. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 845–849, Shenzhen, China, December 15-18 2016.

5. Sara Santiso, Arantza Casillas, Alicia Pérez, and Maite Oronoz. Medical entity recognition and negation extraction: Assessment of NegEx on Health Records in Spanish. In *2017 International Work-Conference on Bioinformatics and Biomedical Engineering (IWBBIO)*, pages 177–188, Granada, Spain, April 26-28 2017.

6. Sara Santiso, Arantza Casillas, and Alicia Pérez. The class imbalance problem detecting Adverse Drug Reactions in Electronic Health Records. *Health Informatics Journal*, 1–11, 2018.

7. Sara Santiso, Arantza Casillas, Alicia Pérez, and Maite Oronoz. Word embeddings for negation detection in health records written in Spanish. *Soft Computing*, 1–7, 2018.

5

Adverse Drug Reaction detection with dense representations and Random Forest

5.1 Introduction

The characterization of the drug-disease pairs is an important factor in the ADR detection from EHRs. We observed that the results obtained by the symbolic characterizations were not good enough for the ADR detection system. The underlying difficulties from the symbolic characterization are the lack of corpus and the lexical variability. In Chapter 3 we explained that EHRs are subject to strict confidentiality regulations making the access to them difficult and, as a consequence, there are few examples to train the predictive model. Furthermore, the same concept can be expressed by different word-forms, with or without abbreviations, using either standard or non-standard terminology and with misspellings. For example, the term 'diabetes mellitus tipo 2' was also written as 'diabetes mellitus tipo II', 'diabetes tipo 2', 'dm tipo 2', or 'dm2'. Accordingly, machine learning from text with disperse symbolic representations is not straightforward (Farhan et al., 2016).

Then, we found necessary representations robust against lexical variations and we decided to characterize the drug-disease pairs using dense representations based on embeddings. The embeddings (Mikolov et al., 2013a) are vector representations of the words in a continuous space of small dimension. That is, each word is mapped to a vector of real numbers enabling, thus, algebraic operations. These dense or continuous representations of words

help inference algorithms to achieve a better performance in NLP tasks by grouping semantically related words (Gormley et al., 2015). The main benefit of the embeddings is the generalization ability (Goldberg and Hirst, 2017). In ADR extraction we can find related works that included embeddings in their representations. Nikfarjam et al. (2015) and Lin et al. (2015) combined discrete features, such as the context, the POS or the lemmas, with the clusters obtained from the embeddings. Given that both works treated the ADR extraction as a mention detection task, they used the CRF classifier, frequently used in NER tasks. Henriksson et al. (2015a) used entity classes, entity unigrams, entity bigrams, distance and context represented by words, semantic vectors and multiple semantic vectors. Furthermore, they had to apply sub-sampling and weights to overcome the class imbalance of ratio 1:8. Henriksson et al. (2015b) made ensembles of different models created changing the window size of the context used to generate the embeddings. In both cases they took into account the drug and the disease and they used the RF classifier. Zhang et al. (2016) used word-embeddings by averaging the vectors of each tweet. These embeddings were used in one of the classifiers of their ensemble method implemented with ME. These works concluded that the dense representations based on embeddings improved the results.

Despite exploring dense representations, given that we do not have corpora as large as in other domains, we had doubts concerning the quality of our generated embeddings. Then, our hypothesis was that our embeddings might be not good enough and we transformed the continuous space into a coarse-grained one. In this way, these smoothed space would enable to avoid superficial variations derived from errors.

It is important to explain that, according to the findings from Chapter 4, we decided to focus on intra-sentence ADRs. That is to say, we only found ADRs that have the drug and the disease in the same sentence. Furthermore, we continued using the RF algorithm to infer the model, as was done with the symbolic representation. This allowed us to compare the impact of the representation with the same classifier. We also employed the gold mentions (the entities manually annotated by the experts) to focus on the ADR detection.

With the experiments developed in this chapter, our aim is to address the following research questions:

Research Question 3

Can dense features be used to represent ADRs in order to help to overcome the lexical variability of the EHRs written in Spanish?

Research Question 4

Given that dense spaces might be unreliable because the corpora formed by EHRs tend to be small, is it advisable to transform the original dense spaces into coarse-grained ones using smoothing techniques?

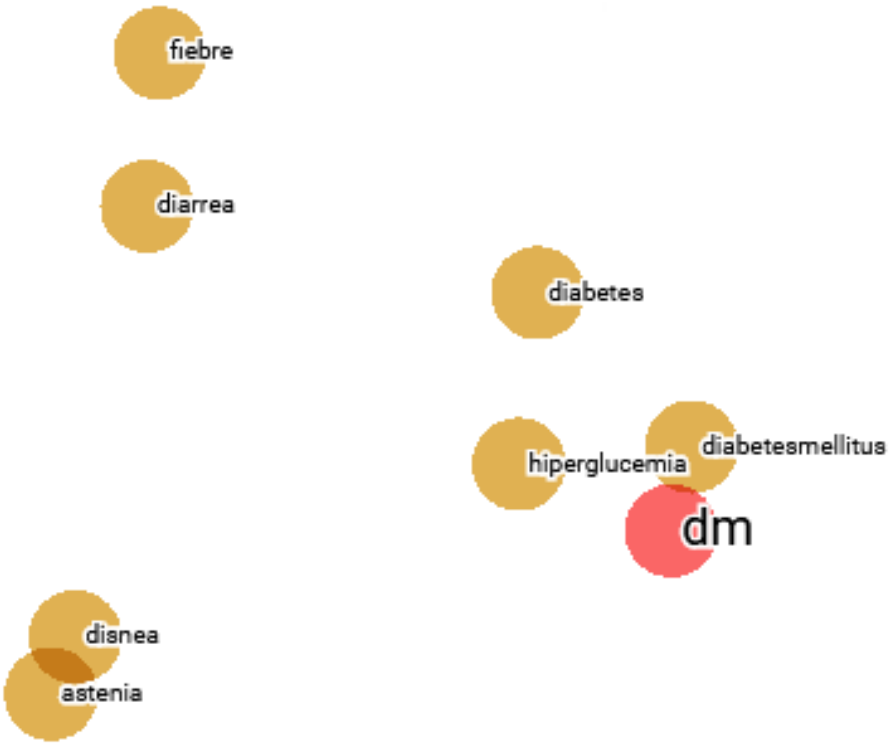
The rest of the chapter is organized as follows: Section 5.2 explains the embeddings used for the dense representation, generated with different approaches and corpora. Section 5.3 explains simple smoothing techniques proposed to smooth the dense space. Section 5.4 describes the features employed to characterize the ADRs. Section 5.5 gives the results obtained during the evaluation of the different experiments and the discussion of them. Section 5.6 provides the final conclusions.

5.2 Word-embedding generation

Given that we decided to use a dense representation to characterize the ADRs, we had to select the techniques and the corpora to create the embeddings. The embeddings were trained with unsupervised methods and are based on the hypothesis that words with similar meanings tend to appear in similar contexts (Harris, 1954). These methods have as input an unannotated corpus and the output is a vocabulary (Σ) where each word of the corpus appears together with the corresponding vector in $\mathbb{R}^n$, as shown in Figure 5.1a. Figure 5.1b shows the representation of some example embeddings in the vector space. We can see that words with similar meaning are close to each other. For example, the word 'dm' is closer to '*diabetesmellitus*', '*diabetes*' and '*hiperglucemia*' than to '*disnea*', '*fiebre*', '*diarrea*' and '*astenia*'. Note that this example also makes possible to observe that the lexical variants of the same concept appears close to each other. The words '*diabetes*', '*diabetesmellitus*' and '*dm*' refer to the same disease and appear close in the vector space.

$w \in \Sigma$	$f(w) \in \mathbb{R}^n$				
diabetesmellitus	-0.043090	-0.020025	...	-0.029354	-0.014414
diabetes	0.171899	-0.216604	...	-0.118141	0.150151
dm	-0.073951	-0.006072	...	-0.146678	0.008422
hiperglucemia	-0.129522	0.046963	...	0.069404	0.025743
disnea	-0.093281	0.054698	...	-0.199195	0.254324
astenia	-0.345448	0.156622	...	-0.208091	0.042582
diarrea	-0.167453	0.097724	...	-0.113169	0.034976
fiebre	0.043911	-0.008858	...	-0.151012	-0.198613

(a) Vocabulary obtained in the embedding generation where each word appears together with the corresponding embedding.



(b) Projection of the embeddings in the vector space.

Table 5.1: Example of embeddings obtained with the embedding generation approaches. These embeddings correspond to the words '*diabetesmellitus*', '*diabetes*', '*dm*', '*hiperglucemia*', '*disnea*', '*astenia*', '*diarrea*', '*fiebre*' (meaning 'diabetesmellitus', 'diabetes', 'dm', 'hyperglycemia', 'dyspnea', 'asthenia', 'diarrhoea' and 'fever' respectively).

The **embedding extraction approaches** selected for this work were the following ones:

- **word2vec**: We used the Continuous Bag-of-Words (CBOW) architecture which projects all the words in the same position by averaging the vectors of the context. That is, the word is predicted from the context. In this case, the order of the context words does not matter (Mikolov et al., 2013a).

- **skipNgram**: We used the Skip-Gram architecture which tries to maximize classification of a word based on another word in the same sentence. That is, the context is predicted from the word. In this variant the model is sensitive to the positioning of the words (Ling et al., 2015).

- **GloVe**: It is trained on a global word-word co-occurrence matrix, which contains the statistics of how frequently words co-occur with one another in a given corpus. It tries to minimize the sum of the squares of a log-bilinear model (Pennington et al., 2014).

These models have in common that are inspired by neural networks. However, GloVe is based on the co-occurrence of word pairs and word2vec with CBOW and skipNgram are based in the context. Apart from the aforementioned approaches, there are more recent approaches such as FastText (Bojanowski et al., 2017), which is an extension of Skip-Gram that takes into account subword information, ELMo (Peters et al., 2018), which generates context-sensitive representations from language models, or BERT (Devlin et al., 2018), which context-sensitive representations from masked language models in order to take into account the left and right contexts simultaneously. Unfortunately, when we developed this task these approaches were not available. Then, we did not include these in our experimentation.

Regarding the **corpora**, to represent the words into a dense space, a large unannotated corpus is required. In this work we employed embeddings extracted from an **in-domain** corpus formed by EHRs (denoted as uEHR). We also explored an **out-domain** corpus formed by SBWCE. Both are written in Spanish, but the number of words of the out-domain dataset is approximately 10 times higher than in the in-domain dataset (109,618,393 vs 1,420,665,810) and their vocabulary is 4 times higher (286,984 vs 1,000,653) (turn to Section 3.2.2 for more details about the unannotated corpora). The motivation was to asses the impact in the lexical variability and, specifically, in the OOV words.

5.3 Smoothing techniques

To learn high-quality word-embeddings, huge datasets are needed (Mikolov et al., 2013a,b). We were aware of the fact that datasets on EHRs tend to be sparse. Hence, we assumed that the vectors derived from the word-embedding generation process would not be ideal. If the corpus is small, the frequency of the words is low and it is difficult for the embeddings to establish the semantic relationships. This can be seen with the in-domain and the out-domain datasets, the ratio between the number of tokens and the vocabulary is about 1:300 for the in-domain and 1:1,000 for the out-domain. That is, in the in-domain corpus the words are repeated with a lower frequency.

As a consequence, we decided to apply smoothing techniques to transform the representation space into a more coarse-grained one as if we had zoomed-out. In other application context, smoothing techniques enabled to extract more flexible and robust information from data by capturing patterns and avoiding noise (Simonoff, 2012). It can seem contradictory turning to a continuous representation to, next, obtain a more coarse-grained representation. However, the vectors obtained with smoothing techniques let us to benefit from the dense representations and, at the same time, improve the proximity between semantic related words that was lost due to the few corpus available to train the embeddings.

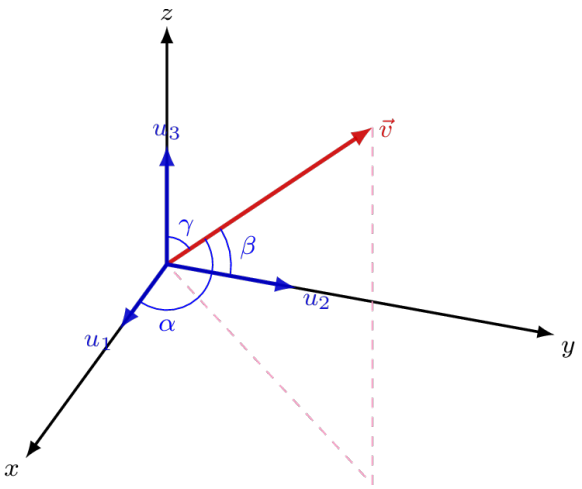
Next, we explain the four smoothing techniques used in this work: i) direction cosines, ii) truncation, iii) Principal Component Analysis (PCA), and iv) clustering.

5.3.1 Direction cosines

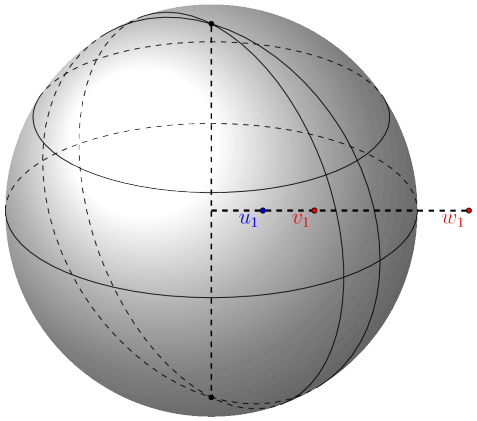
One of the smoothing techniques examined in this work is the transformation of the algebraic basis vectors into the representation space provided by **direction cosines**, also known as length normalization (Artetxe et al., 2016). The direction cosines are the cosine of the angles that the vector forms with the coordinate axes, each of which serves as a component of the direction vector. Given a vector $\vec{v} = (x_1, \cdots, x_n) \in \mathbb{R}^n$, its corresponding direction vector $\vec{u}$ is obtained as in expression (5.1), being $||\vec{v}||$ the norm of $\vec{v}$. It is straightforward to verify that $||\vec{u}|| = 1$. That is, the direction cosines are normal and, in spheric coordinates, all the elements would have a radius $r = 1$. In the three-dimensional ($n = 3$) space, given $\vec{v}$ we would get, as direction cosines, $\vec{u} = (u_1, u_2, u_3) = (cos\alpha, cos\beta, cos\gamma)$ with the angles shown

in Figure 5.1.

$$\vec{u} = (u_1, \cdots, u_n) = \frac{\vec{v}}{||\vec{v}||} = \left(\frac{x_1}{||\vec{v}||}, \cdots, \frac{x_n}{||\vec{v}||} \right) \tag{5.1}$$



(a) The direction cosines associated to vector $\vec{v}$ are (u_1, u_2, u_3), that correspond, respectively, to the cosine of the angles α, β, γ.



(b) The points v_1 and w_1 are represented by u_1 after applying of the direction cosines.

Figure 5.1: Smoothing with direction cosines settles equivalences between two vectors in the same direction and different radius.

5.3.2 Truncation

Another technique used to smooth a continuous space is **truncation**. Truncating by a second or third significant digit regards all the elements in a small neighborhood as equivalent. As depicted in Figure 5.2, all the elements in a small hyper-cube are assimilated by the same vector. This technique was employed in the medical domain by Henriksson et al. (2015a). However, they rounded up the cosine similarity used as feature in their representation instead of the components of the vectorial representation.

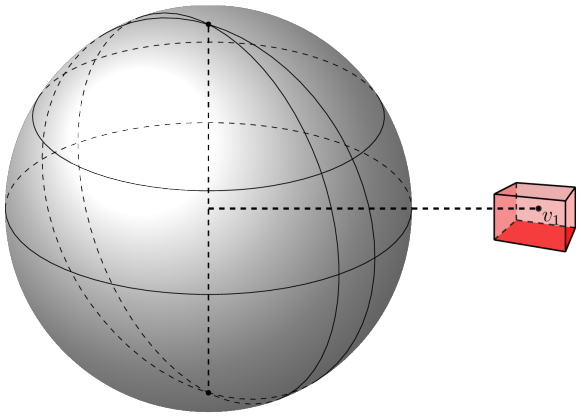


Figure 5.2: Smoothing by truncating makes equivalent elements close in a small hyper-cube. Al the points in the hyper-cube, such as v_1, are represented by the same vector after applying of truncation.

5.3.3 Principal Component Analysis

We employed **PCA** to reduce the dimension of a continuous space. The original vectors may have correlated variables that are transformed into linearly uncorrelated variables called principal components. As a result, the basis is changed into a smaller set of orthogonal components in the directions of maximum variance. PCA was used in the medical domain by Jacobson and Dalianis (2016) to reduce the dimension of the vectors. Unlike our work, they applied PCA to TF-IDF values and not to embeddings.

5.3.4 Clustering

We used **clustering** as a natural means of decreasing the granularity of a continuous space. Clustering creates equivalent classes in a way that close points in the continuous space are assigned a label from a finite set. This can be seen as an alternative means of discretization. Clustering embeddings was used in other works related with the medical domain (Nikfarjam et al., 2015; Lin et al., 2015) to add a high level abstraction to the feature space by assigning the same cluster to similar tokens. In both cases the clusters were included in symbolic representations and not in dense representations. In our work we proposed to cluster the entity vectors using two approaches:

- **K-means**: We used the k-means algorithm implemented in word2vec. This assigns the vector to the cluster with the nearest centroid, which consists in the mean of the vectors of the cluster and is initialized by assigning randomly a vector (Mikolov et al., 2013a).
- **Brown**: It is a hierarchical clustering where a greedy algorithm merges those clusters for which the loss in average mutual information is least (Brown et al., 1992). In this work, the binary paths that represent the Brown clusters were truncated to reduce the granularity. That is to say, the three was pruned.

5.4 Dense characterization

The different embedding generation approaches and smoothing techniques gave place to multiple dense representations that were used during the experimentation. In this section we describe these representations. Note that we continued addressing the ADR extraction as a relation extraction task, contrary to Lin et al. (2015), Nikfarjam et al. (2015), or Henriksson et al. (2015b), to take into account the drug and the disease involved in the ADR. The characterization of the drug-disease pairs was based on the characterization of each entity, denoted as x and y.

To develop the experimentation, we found necessary to have a symbolic representation that could be used to make a direct comparison with a basic dense representation. To this end, we created two equivalent representations, where the words were represented with symbolic features in one case and with dense features in the other case. We want to remark that this symbolic representation is not the one used in Chapter 4 because it contains more features.

As a result, we have three perspectives described below: i) concatenation of words, ii) concatenation of embeddings, and iii) context-aware embeddings.

First, we used a symbolic representation denoted as **concatenation of words** (**Baseline-0**). This leads to a nominal space that encompasses the word-forms within the left and right contexts of each entity. Second, we used a characterization where each aforementioned word-form was replaced by its corresponding embedding, denoted as **concatenation of embeddings** (**Baseline-1**). This is a straightforward variant to convert the symbolic space into a dense space. Third, we used the **context-aware embeddings** to represent each entity and their contexts, which were computed as in (5.2). This performs a weighted sum of the embeddings in a context-window of length k with respect to the focus word, the target entity $\mathbf{v}_j$, and yields $\tilde{\mathbf{v}}_j$. This context-aware representation is much more compact than the concatenation of embeddings, since it keeps the dimension of the search space rather than increasing it by concatenating all vectors. Another interesting fact about the context-aware representation is that the weight (w_i) serves to either diminish or enhance the importance of the context-words.

$$\tilde{\mathbf{v}}_j = ca(\mathbf{v}_j) = \mathbf{v}_j + \sum_{\substack{-k \leq i \leq k \\ i \neq 0}} w_i \mathbf{v}_{i+j} \tag{5.2}$$

In this work we explored three **weighting strategies**: i) constant weight: $w_i = 1$, ii) diminishing the contribution of context words as the distance to the drug or disease increases: $w_i = \frac{|k-i+1|}{2k}$ (the weight is proportional to the distance between the context-word and the entity in the total number of context-words), and iii) learning the weights of each word according to the Information Gain (Quinlan, 1986): $w_i = InfoGain(c, s_{j+i})$, being c the class and s_{j+i} the symbolic representation that corresponds to the vector $\tilde{\mathbf{v}}_{j+i}$.

We would like to point out that in order to obtain the embedding of a token with more than one word, we computed the centroid (average of the vectors involved). In this way, external resources are not needed and this arithmetic can be used for embeddings of any domain. For example, the disease s '*descompensación cardiaca izquierda*' (meaning 'left heart failure') comprises three words: s_1 = '*descompensación*', s_2 = '*cardiaca*' and s_1 = '*izquierda*', that correspond to three vectors: $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ respectively. Then, s is represented by $\mathbf{v} = \frac{\mathbf{v}_1+\mathbf{v}_2+\mathbf{v}_3}{3}$. The OOV words were represented by the null vector ($\overrightarrow{\mathbf{0}}$). In our case, with the arithmetics involved to compute the vectors described in (5.2), it works as if we had omitted the OOVs.

Finally, we made use of additional features related with the vector of each entity: the modulus of the disease and drug vectors ($|\mathbf{x}|$ and $|\mathbf{y}|$). We also made use of features associated with the relation vector $\overrightarrow{\mathbf{xy}}$: the modulus ($|\mathbf{xy}|$) and the cosine of the angle between the entity vectors ($\cos\theta$).

Altogether, the different ADR representations explored in this work are summarized in Table 5.2. In addition, these features are shown graphically in Figure 5.3 for the drug-disease pair '*esteroideo - descompensación hiperglucémica*' (meaning 'steroidal - hyperglycaemic decompensation'), which was extracted from the sentence '*A consecuencia del tratamiento esteroideo se produce descompensación hiperglucémica que precisa tratamiento con insulinización*' (meaning 'As a result of the steroidal treatment, it was produce an hyperglycaemic decompensation that needs treatment with insulinization'). Note that we organized the experiments in a way that we could see the influence of each feature and smoothing technique.

First, we used the word-based representation of each entity with their contexts (referred to as representation 0) to develop Baseline-0. After that, we replaced the words by their corresponding embeddings (representation 1) to develop Baseline-1, which allows to compare symbolic and dense characterizations. These baselines appear under "BASELINE" in Figure 5.3. Next, we turned to the proposed context-aware representations. These ones appear under "CONTEXT-AWARE" in Figure 5.3. We obtain the context-aware representation (representation 2) to compare it with the concatenation of the embeddings used in the previous representation. To the context-aware representation created in the euclidean space we added truncation (representation 3) and PCA (representation 4). Next, we also employed these representations using the direction cosines (representation 5, 6 and 7 respectively) to compare this space with the previously explored euclidean space. Furthermore, we added the Brown and K-means clusters to the previous representation (representation 8). Note that PCA was not applied to the vector of each entity, but to the entire vector of the pair. These representations are grouped under "SMOOTHING" in Figure 5.3. Finally, we included the additional features, that is, the modulus of the drug, the disease and the relation vector together with the cosine of the angle (representation 9 and 10). These representations are grouped under "ADDITIONAL" in Figure 5.3.

Let us now mention the resulting dimension for each representation (see Table 5.2). Baseline-0 involved two entities with a context-window of k words yielding $2k+1$ features for each entity. With $k=3$ we obtained $2\cdot(2k+1) = 14$ symbolic features. Baseline-1 replaced each word by their corresponding

vector of 300 dimensions yielding $2 \cdot (2k+1) \cdot 300 = 4,200$ numeric features. When we used the context-aware representation, each entity was represented by one vector obtaining $2 \cdot 300 = 600$ numeric features. When we applied PCA, we reduced the dimensions of the vector to 50. The clusters just added 1 dimension to the representation. The same happened with the additional features, they added 1 dimension to the representation.

		Features				Smoothing						Additional			
		Space		Entities		Space		T	PCA	Cluster		Modulus		θ	
Approach	Representation	Symbolic	Dense	Concatenation	Context-aware	Euclidean	Direction-cosines	Truncate	PCA($\mathbf{x}, \mathbf{y}$)	K-means	Brown	Entities: $\lvert\mathbf{x}\rvert$, $\lvert\mathbf{y}\rvert$	Relation: $\lvert\overrightarrow{\mathbf{xy}}\rvert$	Relation $\cos\theta$	Dimension
B0	0	✓		✓		✓									14
B1	1		✓	✓		✓									4,200
context-aware	2		✓		✓	✓									600
	3		✓		✓	✓		✓							600
	4		✓		✓	✓		✓	✓						50
	5		✓		✓		✓								600
	6		✓		✓		✓	✓							600
	7		✓		✓		✓	✓	✓						50
	8		✓		✓		✓	✓	✓	✓	✓				54
	9		✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	58
	10		✓		✓		✓	✓	✓			✓	✓	✓	54

Table 5.2: Different dense characterizations to represent ADR relations led us to different experimental approaches (B0 and B1 stand for Baseline-0 and Baseline-1 respectively). Note that "Features" corresponds to the set of words or embedding used to represent the drug-disease pair, "Smoothing" corresponds to the smoothing techniques applied to the context-aware embeddings and "Additional" corresponds to those features derived from the embeddings. The last column shows the dimension of the feature-space.

BASELINE

	A	consecuencia	del	tratamiento	esteroideo
B0		consecuencia	del	tratamiento	esteroideo
B1		*1 ... 300* -0.042554,...,-0.789032	*1 ... 300* -0.255258,...,0.210154	*1 ... 300* 0.228128,...,-0.923926	*1 ... 300* -0.041992,...,0.084206

...

	se	produce	descompensación	hiperglucémica
B0	se	produce	descompensacion-hiperglucemica	
B1	*1 ... 300* -0.041992,...,-0.084206	*1 ... 300* -0.236328,...,-0.117341	*1 ... 300* 0.424497,...,0.228128	

...

	que	precisa	tratamiento	con	insulinización
B0	que	precisa	tratamiento		
B1	*1 ... 300* -0.923926,...,-0.483665	*1 ... 300* 0.424497,..., 0.065513	*1 ... 300* -0.363645,...,-0.236328		

CONTEXT-AWARE

		A	consecuencia	del	tratamiento	esteroideo	se	produce	descompensación	hiperglucémica	que	precisa	tratamiento	con	insulinización
SMOOTHING	**context-aware**		*1 ... 300* -0.873661,...,2.39868							*1 ... 300* -0.885388,...,1.222755					
	+ truncation		*1 ... 300* -0.87,...,2.4							*1 ... 300* -0.89,...,1.22					
	+ PCA					*1 ... 50* 1.587967,...,1.151808									
	direction cosines		*1 ... 300* -0.027779,...,0.07627							*1 ... 300* -0.033909,...,0.046829					
	+ truncation		*1 ... 300* -0.03,...,0.08							*1 ... 300* -0.03,...,0.05					
	+ PCA					*1 ... 50* -3.866848,...,-0.041581									
	clusters					*Brown* 1001010110			*Brown* 101110111						
						K-means 230			*K-means* 466						
ADDITIONAL	**modulus**					$\|x\|$ 31.44999			$\|y\|$ 26.111036						
			$\|\vec{x}\vec{y}\|$ 19.306679												
	θ		$\cos\theta$ 0.745022												

Figure 5.3: Scheme of the features included in each dense representation of the ADR '*esteroideo - descompensación hiperglucémica*'. The features related with the entities are highlighted in dark blue and the features related with the context in light blue.

5.5 Results

In this section we give the results obtained with the aforementioned representations. The experiments were done with the RF classifier (Breiman, 2001), following the same conditions that in the best performing experiment of Chapter 4 (see Section 4.5).

The embeddings of these experiments were trained using a window of size (s) 10 and yielding vectors of 300 dimensions. The window s used to extract the embeddings should not be confused with the context-window k of expression (5.2) used to create the context-aware embeddings. Regarding the practical details of applying smoothing techniques, the components of the vectors used to represent the drug-disease pairs were truncated to 2 decimals. Next, we reduced to 50 dimensions using PCA implemented in Java, including libraries available in Weka (Hall et al., 2009). The PCA dimension was decided on a grid-search. The optimization criterion used the f-measure of the positive class obtained with the hold-out scheme, inferring the model with the train set and evaluating with the dev set. Furthermore, we proposed to cluster the entity vectors with a set of 500 clusters. The Brown clusters were truncated to a maximum of 10 bits. To assess the models we also used the IxaMed-GS corpus and the hold-out evaluation scheme, as it was explained in Section 3.3.1.

First, we obtained the **baselines** (see Table 5.3) that correspond to representations 0-1 in Table 5.2. The results corroborated that the dense representations outperformed the symbolic representation, except for the embeddings trained with GloVe and the in-domain corpus. We concluded that performance with dense characterizations, even used in replacement of word-forms, is much better than with symbolic characterizations. In our best case, the f-measure of the positive class improved from 36.8 to 51.7. For the out-domain corpus we generated the embeddings with skipNgram given that this architecture is better at capturing infrequent words. Surprisingly, these out-domain embeddings offered good results, being the second best result among the baselines with an f-measure for the positive class of 46.2. Despite this, word-embeddings can be used in a smarter way with the context-aware representations.

	Space		Embedding Extraction			Corpora		Evaluation			
Approach	Symbolic	Dense	Word2vec	SkipNgram	GloVe	In-domain	Out-domain	Precision	Recall	F-measure	Class
B0	✓							30.4	46.7	36.8	⊕
								86.4	76.1	81.0	⊖
								76.2	70.7	72.9	W. Avg.
								70.7	70.7	70.7	Micro Avg.
								58.4	61.4	58.9	Macro Avg.
B1		✓	✓			✓		53.6	50.0	51.7	⊕
								89.0	90.3	89.6	⊖
								82.5	82.9	82.7	W. Avg.
								82.9	82.9	82.9	Micro Avg.
								71.3	70.1	70.7	Macro Avg.
		✓		✓		✓		48.1	43.3	45.6	⊕
								87.6	89.6	88.6	⊖
								80.4	81.1	80.7	W. Avg.
								81.1	81.1	81.1	Micro Avg.
								67.9	66.4	67.1	Macro Avg.
		✓			✓	✓		23.7	30.0	26.5	⊕
								83.3	78.4	80.8	⊖
								72.4	69.5	70.8	W. Avg.
								69.5	69.5	69.5	Micro Avg.
								53.5	54.2	53.6	Macro Avg.
		✓		✓			✓	54.5	40.0	46.2	⊕
								87.3	92.5	89.9	⊖
								81.3	82.9	81.9	W. Avg.
								82.9	82.9	82.9	Micro Avg.
								70.9	66.3	68.0	Macro Avg.

Table 5.3: Baseline results in either a symbolic or a dense space for the dev set of the IxaMed-GS corpus using the Random Forest classifier.

Next, we explored alternative **context-aware** representations, to be precise, representations 2 to 10 in Table 5.2. These experiments yielded more than 30 new results with different metrics each one. Instead of tabulating them all (as we did for the baselines) for compactness, we simply summarized them focusing on the f-measure of the positive class, as shown in Figure 5.4. Turn to Appendix C to see detailed results of these experiments. If we focus on the embedding generation approaches, the embeddings trained with GloVe achieved the best performance using the context-aware representation, obtaining an f-measure for the positive class of 62.3. If we focus on the characterization of ADRs, those representations that applied only truncation (representations 3 and 6) or truncation and PCA (representations 5 and 7) were more effective applied to the direction cosines than applied to the euclidean space. Specifically, using the vectors generated with GloVe and the in-domain corpus (the best performing experiment), the f-measure increased from 49.1 to 53.7 applying truncation to the direction cosines and from 52.2 to 62.3 applying truncation and PCA to the direction cosines. However, the clusters resulted counterproductive, the f-measure decreased from 62.3 to 50.0. Regarding the additional features, these improved the results obtained with the inclusion of the clusters from 50.0 to 60.0, but not the results of the best experiment. All in all, the best representation included three of our four proposals to smooth the vectors of the embedding-based characterization: 1) direction cosines, 2) truncation, and 3) PCA.

Finally, given that the context-aware representations were created with a context-window (k) of length 3, we explored the impact of the length k of the context-window using the constant weighting. In Figure 5.5 we can see that a context-window of size 4 (k=4) improved the f-measure of the positive class obtained with k=3 from 62.3 to 63.9.

Furthermore, we explored the impact of the other two weighting strategies associated with expression (5.2) proposed for the entity characterization described in Section 5.4, that is, diminished weighting and InfoGain weighting. With k=2, the diminished weighting outperformed the other two weights and with k=6, the weight based on InfoGain outperformed the other two weights. However, with k=3 and k=4 the best results were obtained with constant weighting.

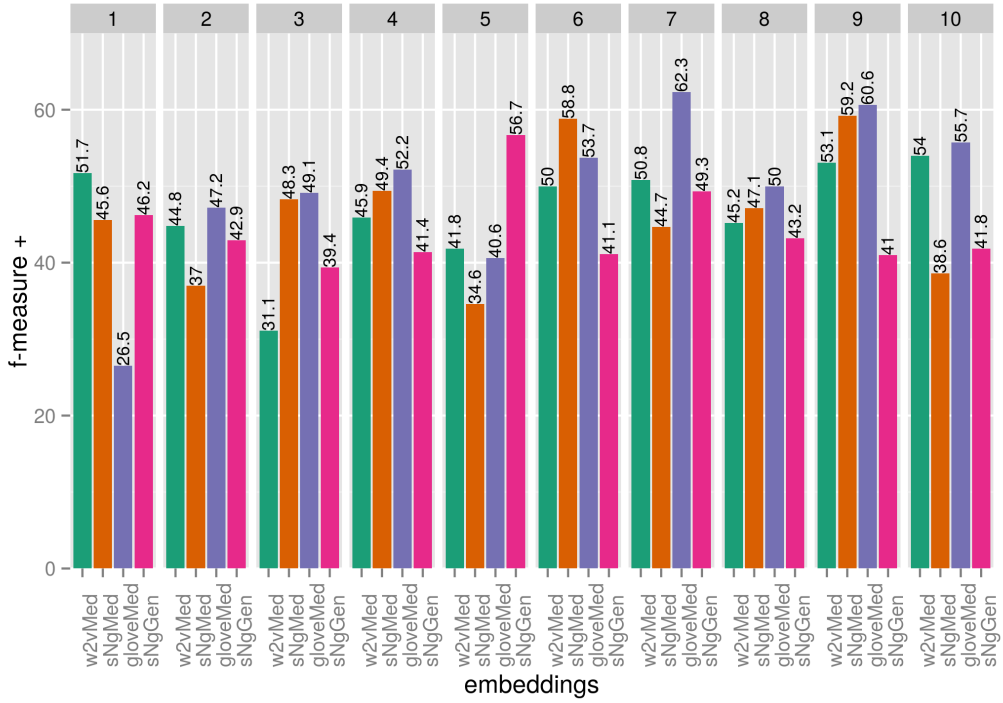


Figure 5.4: F-measure of the positive class with the 10 representations presented in Table 5.2 for the dev set of the IxaMed-GS corpus using the Random Forest classifier. The embeddings were extracted using three different techniques (denoted as w2v, sNg, and glove to refer to word2vec, skipNgram, and GloVe respectively) and from two sources, denoted by the suffix, where Med stands for in-domain medical source and the suffix Gen stands for the general out-domain source.

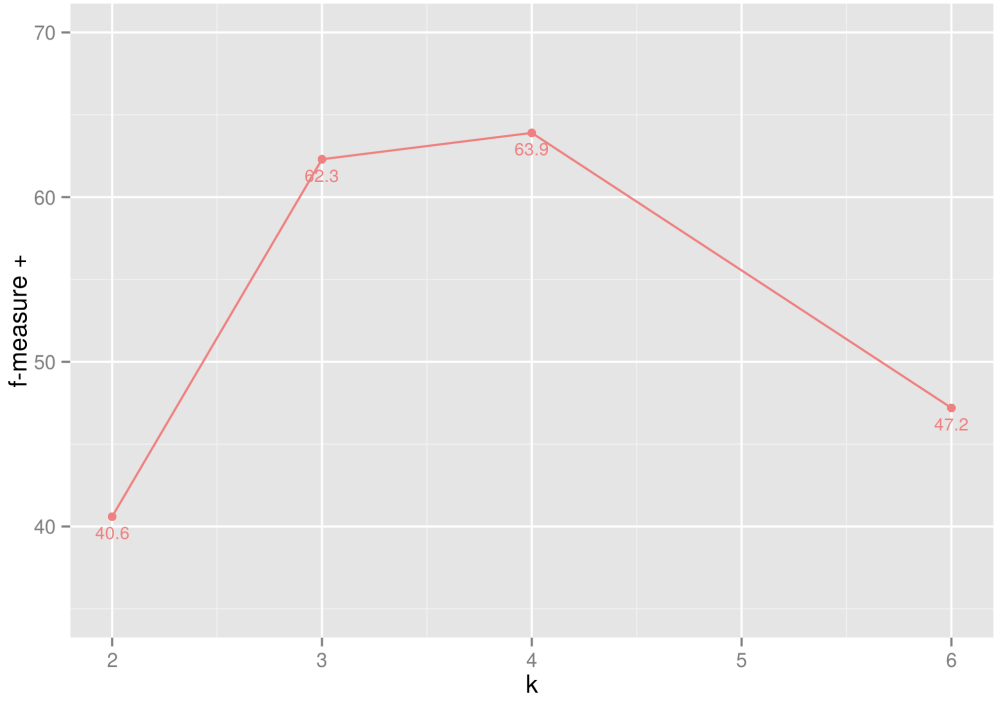


Figure 5.5: F-measure of the positive class varying the length of the context-window (k) for the best performing model (representation 7 in Table 5.2 incorporating the embeddings extracted with GloVe from the in-domain corpus). The model was inferred with the train set and evaluated with the dev set of the IxaMed-GS corpus using the Random Forest classifier. Note that the f-measure of the positive class is represented from 35.

Given that the EHRs tend to have a high lexical variability, an important point to take into account is the tolerance to noise of the model with respect to OOV words. The EHRs of the IxaMed-GS have 37% of instances in which the entities were not found in the embedding vocabulary. To be precise, 22% (71 of 328) of the entities of each drug-disease pair did not have their corresponding embedding. We wondered about the impact of the OOVs on the performance. If we evaluate the model that offered the best performance (representation 7 in Table 5.2) discarding the instances with OOVs from the dev set, we observed a slight improvement in the performance, from 63.9 to 68.3. We also tried to tackle the representation of OOVs by means of character-embeddings, as was proposed by Chen et al. (2015). That is to say, we created the character-embeddings by considering each character as an individual word and using word-embedding generation approaches to learn the character-embeddings. Overall, this approach did not improve the performance of the ADR extraction system. The results of the best performing model decreased from 63.9 to 55.9. So far, we referred to the OOVs derived from the embeddings. For the symbolic representations, there are other type of OOVs mentioned in Chapter 4 (see Section 4.2 and Section 4.5), the words from the dev set that were not found on the train set. In this regard, the IxaMed-GS corpus has 76% of instances with OOVs, that were represented by the missing value. To be precise, 52% (170 of 328) of the entities of each drug-disease pair were not in the vocabulary corresponding to the drugs and the diseases. If we evaluate the model inferred with the symbolic characterization of Baseline-0 (concatenation of words) discarding the instances with OOVs from the dev set, we observed that the results improved from 36.8 to 63.2. The results obtained in these experiments implies that the symbolic characterization produces more OOVs that the dense characterization created with embeddings.

Table 5.4 gives full detailed results achieved with the best performing model (representation 7 in Table 5.2) for the dev and test sets. In both cases the results obtained in Chapter 4 (see Table 4.5) were improved. The f-measure of the positive class increased from 46.2 to 63.9 in the dev set and from 43.2 to 55.4 in the test set.

In addition, we analyzed the ROC curve and the AUC. Figure 5.6 shows the ROC curve and the AUC of the aforementioned experiments for the dev and test sets. In both cases the points are above the diagonal and the AUC obtained in Chapter 4 (see Figure 4.6) was improved, from 0.77 to 0.87 in the dev set and from 0.82 to 0.86 in the test set.

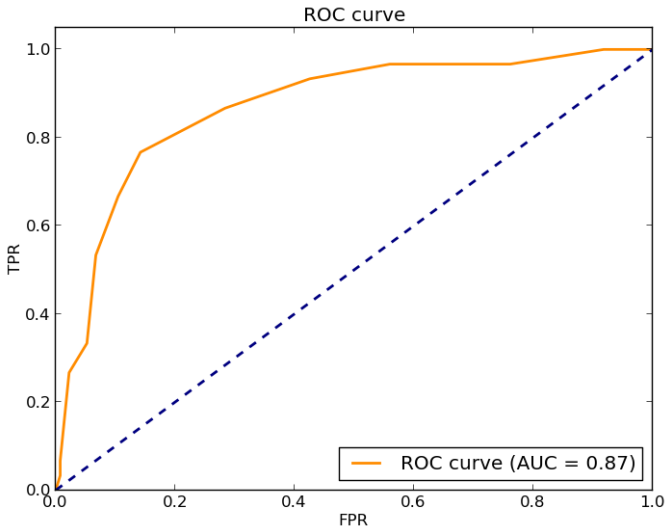
Precision	Recall	F-measure	Class
54.8	76.7	63.9	$\oplus$
94.3	85.8	89.8	$\ominus$
87.0	84.1	85.1	W. Avg.
84.1	84.1	84.1	Micro Avg.
74.5	81.2	76.9	Macro Avg.

(a) Model inferred with the train set and evaluated with the dev set.

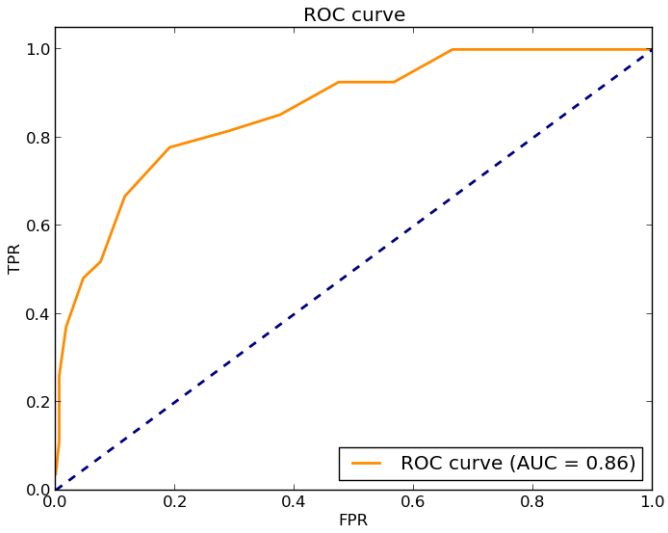
Precision	Recall	F-measure	Class
47.4	66.7	55.4	$\oplus$
94.4	88.4	91.3	$\ominus$
88.1	85.5	86.5	W. Avg.
85.5	85.5	85.5	Micro Avg.
70.9	77.6	73.4	Macro Avg.

(b) Model inferred with the train and dev sets and evaluated with the test set.

Table 5.4: Results of the best performing model (representation 7 in Table 5.2) inferred with the IxaMed-GS corpus and the Random Forest classifier. The dense representation was extracted with GloVe from the in-domain corpus, with embeddings of 300 dimensions and a context-window of size 4.



(a) Model inferred with the train set and evaluated with the dev set.



(b) Model inferred with the train and dev sets and evaluated with the test set.

Figure 5.6: ROC curve and AUC of the best experiment (representation 7 in Table 5.2). The model was inferred with the IxaMed-GS corpus and the Random Forest classifier. The dense representation was extracted with GloVe from the in-domain corpus, using embeddings of 300 dimensions and a context-window of size 4.

5.5.1 Discussion

We compared the results obtained using the **concatenation of words** with those obtained replacing the symbolic features by arrays of numbers, that is, **concatenation of embeddings** (see Table 5.3). Unexpectedly, replacing the words by their embedding led to significant improvements, but the resulting feature vectors had a high dimension. Even though the embedding concatenation turned out to be useful, the continuous space offered alternative ways to encompass the context using a feature vectors of lower dimension. Instead of using an embedding for each word, we built a single **context-aware embedding**. We explored different dense representations by virtue of sets of embeddings extracted from in- and out-domain corpora, as well as different generation techniques (word2vec, skipNgram, GloVe) and dimensions. As was known from previous works (Lai et al., 2016), different embedding generation strategies and settings have an impact on the performance, while no single technique suitable for all domains and tasks has been reported in previous work. In our case, the difference between the out-domain embeddings and the best performing approach was small. Note that the content of the out-domain corpus comprises 13 times more word-forms than the in-domain corpus. Wang et al. (2018b) concluded that embeddings trained with in-domain corpus do not necessarily have better performance than those trained with general-domain corpus. From our results we learned that using a corpora with representative variants of contexts might mitigate the lack of specific corpora.

Given that we were aware of the fact that in-domain corpora tend to be scarce, we did not rely completely on the inferred space and conformed to a bigger-grained space. To do this, we explored simple though efficient **smoothing techniques** applicable on dense spaces. We observed that the smoothing techniques outperformed their corresponding non-smoothed counterpart. Note that ADRs are rare cases, hence, in the training process the ADRs are under-represented and, accordingly, the statistics obtained might be biased to the majority cases. Thus, smoothing helped to the location of rare cases in a nearby region. In particular, truncating the space is equivalent to a discretization and, in this task, it shown significant benefits to the classification of ADRs, and even more so if the representation space chosen is given in terms of direction cosines. In general, the dimension reduction with PCA also resulted beneficial. In fact, in the experiment with the best results we made use of PCA and this finding is important because using fewer

features speeds up the training process. Unfortunately, the clusters used in other works with promising results did not benefit our proposed ADR characterization and were discarded.

We also observed that taking a context-window of 4 words with equally weighting was reasonable for ADR detection. The other two weightings (diminishing weighting and InfoGain weighting) did not show improvements.

Regarding the **OOV** words, the embeddings were less sensitive to OOVs than the symbolic representations because fewer instances were affected. As consequence, there are smaller variations in the results when the instances with OOVs are discarded, particularly, in comparison to the impact of OOVs with symbolic features. We found that other possible option to tackle the OOVs could be to use FastText since it generates the embeddings of subwords units. Furthermore, the use of a context-aware representations and the smoothing techniques, principally PCA, helped to avoid the OOV words.

5.5.2 Error analysis

After analyzing the experimental results, we compared the predictions made by the best performing model, obtained with the smoothed dense representation and re-sample, and the predictions made with the symbolic representation and re-sample, the best performing model of Chapter 4.

We found that, with the dense representation, the ADRs of the example given for the symbolic one were also detected. In the sentence shown in Figure 5.7 we can see that the pairs '*hipoglucemia - septrin*' and '*hipoglucemia - timetropin*' were detected (the pairs '*disminución del apetito - septrin*' and '*disminución del apetito - trimetropin*' were incorrectly detected as ADRs, one less than with the symbolic characterization). Note that the black arrows "Causada_por" correspond to the ADRs annotated by the experts and the red arrows "Causada_por_system" correspond to the predictions made by the system.

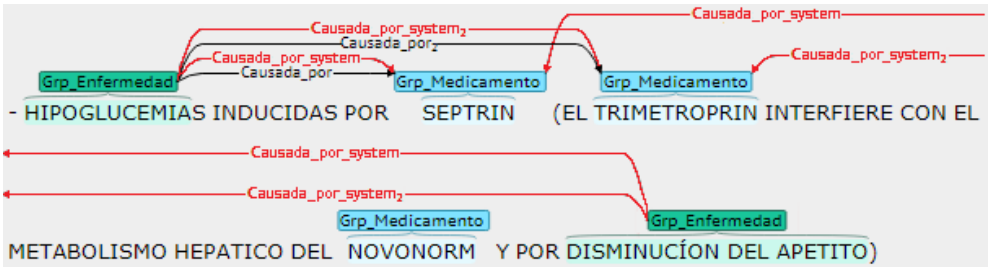


Figure 5.7: Example of sentence in which the model inferred with a dense representation with smoothing detected ADRs discovered by the symbolic characterization. The sentence means 'Hypoglycemias induced by septrin (the trimethoprim interferes with the hepatic metabolism of the novonorm and by decreased appetite)'.

We also found that this dense representation, that incorporates the smoothing techniques, was able to detect ADRs not discovered by the symbolic one. For example, in the sentence shown in Figure 5.8 the ADR '*episodio alérgico - contraste iodado'* was detected (contrary to the happened without applying smoothing), despite of the fact that the ADR '*alérgico - yodo*' appeared for training. Moreover, this ADR was detected in the same way that '*reacción alérgica - contraste iodado*' was discovered previously.

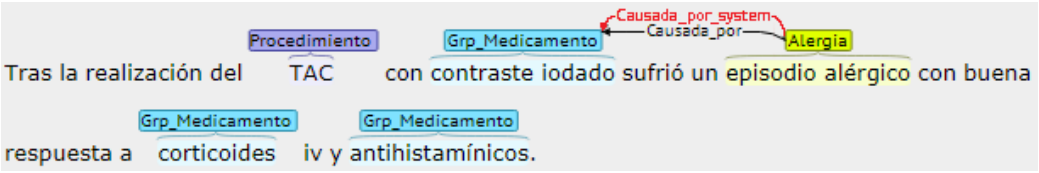


Figure 5.8: Example of a sentence in which the model inferred with a dense representation with smoothing detected correctly the ADR annotated by the experts. This ADR was not detected by the model inferred without smoothing. The sentence means 'After the realization of the CT scan with iodine contrast he suffered an allergic attack with a good response to iv corticoids and antihistamines.'.

This manual analysis also revealed several sources of errors. First, in very **long sentences** (around 40 words) with a wide combination of potential events (around 13), it was not infrequent to find FPs. Second, the system continued detecting as ADR some drug-disease pairs that are related as **treatment**, yielding FPs.

Apart from these errors, some FPs were obtained due to errors in the labels of the entities assigned by the **experts**. This confirms again that gold standards are not necessarily free from errors (Perotte et al., 2014). For example, in the sentence shown in Figure 5.9 the entity "vómito" would be "Grp_Enfermedad" instead of "Grp_Medicamento". This caused of 3 of the 4 FPs: '*hipoglucemias - vómito*', '*nauseas - vómito*', '*disminución del apetito - vómito*'.

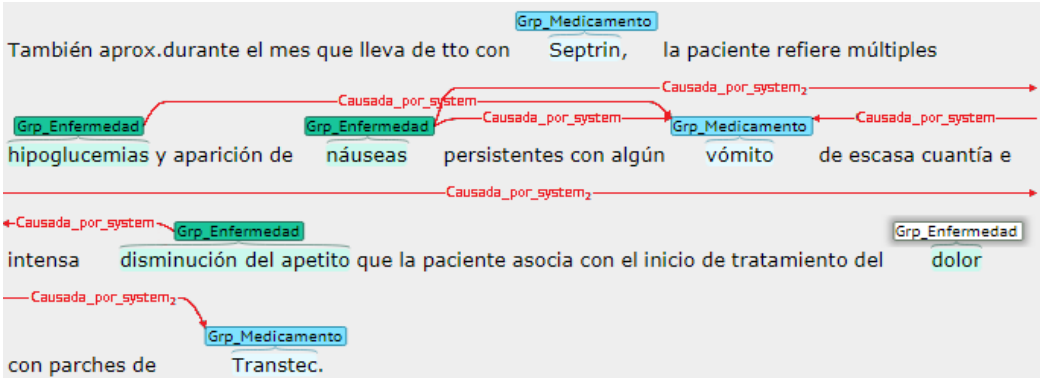


Figure 5.9: Example of sentence in which the best performing model committed 3 of the 4 FPs because some entities were incorrectly labeled by the experts. The sentence means 'Approximately during the month that she has been with the treatment with Septrin, the patient also refers multiple hypoglycemias and appearance of persistent nauseas with some vomit of small amount and intense decreased appetite that the patient associates with the start of the treatment of pain with patches of Transtec.'.

In addition, we observed again some FNs in speculative sentences with medical **uncertainty** (Velupillai and Kvist, 2012), that is, sentences were the diagnosis of the doctor about an ADR is uncertain (this type of error could happen because the system generalizes as non-ADR some uncertain ADRs that were labeled by the experts).

In brief, with the smoothed dense characterization the system was able to detect more ADRs. Some of them corresponded to pairs that were not seen during training because they appeared represented with different terms. We found as sources of FPs long sentences with a wide combination of drug-disease pairs and sentences with drug-disease pairs that are related as treatment. Furthermore, some FPs occurred because of errors in the annotation of the entities made by the experts. The sources of FNs were speculative sentences.

5.6 Conclusions

5.6.1 Concluding remarks

In this chapter we explored dense representation to characterize the ADRs with the aim of overcoming the lexical variability of the EHRs. These achieved better results that the symbolic characterization used in Chapter 4. In practice, we explored the context-aware representations in different continuous spaces generated with in- and out-domain corpora, resulting more efficient the in-domain corpus. Furthermore, we delved into continuous ADR representations combined with simple efficient smoothing techniques. Some of these alternative smoothing techniques resulted useful such as: i) direction cosines: the vector was represented in a different based, ii) truncation: the space was discretized, iii) PCA to carry out dimension reduction.

According to these results, we answered to the following research questions:

Research Question 3

Can dense features be used to represent ADRs in order to help to overcome the lexical variability of the EHRs written in Spanish?

Detecting ADRs from real EHRs is a challenging task due to the high lexical variability. The dense representations created with embeddings enables the machine learning algorithm to perform better than the symbolic characterizations. This could mean that, with dense representations the model is able to generalize to ADRs whose entities are semantically related. For example, terms that correspond to the same disease.

Research Question 4

Given that dense spaces might be unreliable because the corpora formed by EHRs tend to be small, is it advisable to transform the original dense spaces into coarse-grained ones using smoothing techniques?

The use of smoothing techniques such as the combination of direction cosines, truncation and PCA improve the dense representation of the drug-disease pairs. Smoothing helps to avoid superficial variations and, hence, makes different (but close) points in the space equivalent. In this way, the system increases the generalization ability and is able to group diseases or drugs that correspond to the same category.

Open questions. In the symbolic representations as well as in the dense representations we took into account the context by including the surrounding words of the entities in the features. In the case of the dense representations, we also employed context-aware embeddings. Note that we could turn to approach such as ELMo or BERT to generate contextual embeddings. In both cases we had to develop an extensive feature engineering to obtain the ADR characterization. However, we considered that there was still a gap for improvement in making robust the representation of the entities with their context. In this way, our purposes are exploring latent features discovered automatically and generate synergies between the representation and the training of the classifier, because, so far, they were developed as independent processes. Thus, it seems appropriate to employ neural networks for ADR detection in the following chapter.

5.6.2 Publications

This work lead to the following publication:

1. Sara Santiso, Alicia Pérez, and Arantza Casillas. Smoothing dense spaces for improved relation extraction between drugs and adverse reactions. *International Journal of Medical Informatics*.
[accepted, awaiting publication]

6 Adverse Drug Reaction detection with dense representations and Joint Attentive Bidirectional Long Short-Term Memory

6.1 Introduction

For ADR detection, the representation seems crucial. As a consequence, we explored symbolic and dense features, which were used together with the traditional classifier RF. In order to obtain these features, we had to create a tough preprocessing in each case. We also observed that the use of context is important in the representation, but we only considered the contextual information by including in the representation the surrounding words of the entities involved in the ADRs. Furthermore, given that the features and the training of the classifier are developed independently, the features may not be optimized for the classifier.

To overcome these issues we opted to infer the predictive model using deep learning algorithms, where the representation and the model are optimized jointly during the inference. Deep neural networks obtain high-level features using multiple levels of representation obtained by successive transformations, starting from a raw input to more and more abstract levels (Lecun et al., 2015). These algorithms could be useful mainly for two reasons: i) they reduce the need of designing hand-crafted features and ii) there are some architectures that consider the context during the inference. Emerg-

ing trends in classification promote the use of neural networks. The neural networks with embedding-based features outperformed the traditional classifiers with hand-crafted features in several works (Nguyen and Grishman, 2015; Huynh et al., 2016; Feng et al., 2016).

In related works we observed that the authors turned to deep learning for ADR extraction, instead of using the traditional machine learning algorithms. For example, there are works in which the authors employed CNNs such as Lee et al. (2017), that used as core-features phrase embeddings obtained with other CNN. Akhtyamova et al. (2017) and Masino et al. (2018) used word-embeddings as core-features. In other works the authors turned to RNNs, such as Cocos et al. (2017), also with word-embeddings as core-features. Others opted for Bi-LSTM networks such as Gupta et al. (2018) and Stanovsky et al. (2017). The later augmented the word-embeddings with knowledge graph embeddings of DBpedia. Tutubalina and Nikolenko (2017) and Wunnava et al. (2018) combined a Bi-LSTM network and a CRF classifier. The latter augmented the word-embeddings with character-level representations. There are other authors such as Huynh et al. (2016) that explored a CNN and a Recurrent Convolutional Neural Network and proposed two new neural networks, a Convolutional Recurrent Neural Network and a Convolutional Neural Network with Attention. They used GRU in the recurrent layers and observed that the CNN outperformed the rest.

Neural networks learn the representation of words as vectors as part of the training process and also learn to combine word vectors in a way that is useful for prediction (Goldberg and Hirst, 2017). In this way, the time needed to design the hand-crafted features is reduced and the resulting vectorial representation can be helpful to generalize and tackle the data sparsity (Zhou et al., 2016). In this regard, we proposed to employ a lemmatized version of the corpus and the embeddings to create the core-features of the neural networks. The intuition is that the embedded lemmas can help to tackle the variability of the terminology employed by the doctors in the EHRs. Indeed, embedded lemmas were used in other works (Straková et al., 2016) for NER.

Among the different architectures used to implement the neural networks, we can distinguish mainly CNNs and RNNs. Unlike CNNs, RNNs (Elman, 1990) consider the previous states and this allows representing the words in a sequence following the structure of the sentence. The fact of considering the previous states would let us incorporate the contextual information in the representation. For this reason, we opted for LSTM networks (Hochreiter and Schmidhuber, 1997), a type of RNN that uses memory cells to capture

long-range dependencies.

In addition to this, our purpose also was to explore if the neural networks are robust against the class imbalance problem that present our ADR candidates.

At this point, we would like to clarify that we continued using the gold mentions (the entities manually annotated by the experts) to create the drug-disease pairs, that is, our ADR candidates. We also restricted to the intra-sentence ADRs following the findings from Chapter 4.

According to this, in this chapter the aim is to address the following research questions:

Research Question 5

Given that the dense features used to characterize the ADRs are inferred together with the model, can the Bi-LSTM networks help to cope with the lexical variability?

Research Question 6

Are the Bi-LSTM networks sensitive to the class imbalance present in ADR detection?

The rest of the chapter is organized as follows: Section 6.2 describes the neural network architecture and its training process. Section 6.3 gives the results obtained during the evaluation of the different experiments and the discussion of them. Section 6.4 provides the conclusions.

6.2 Joint AB-LSTM

From the previous works we learned that CNNs (Lee et al., 2017; Akhtyamova et al., 2017; Masino et al., 2018; He et al., 2019), RNNs (Cocos et al., 2017), LSTM networks (Luo, 2017) or Bi-LSTM networks (Stanovsky et al., 2017; Gupta et al., 2018; Wunnava et al., 2018; Jagannatha and Yu, 2016b; Li et al., 2017) were used as deep learning algorithms to extract ADRs (defined as presence, mention or relation).

In this work we focused on Bi-LSTM networks since, not only they consider the information of the backward states, but they also use the input sequence in reverse to get information of the forward states. Then, they make possible to infer features containing information about the context of

the entire sentence. Figure 6.1 shows the architecture of a Bi-LSTM where x_1, x_2 and x_3 are the input sequences and z_1, z_2 and z_3 are the outputs of the network.

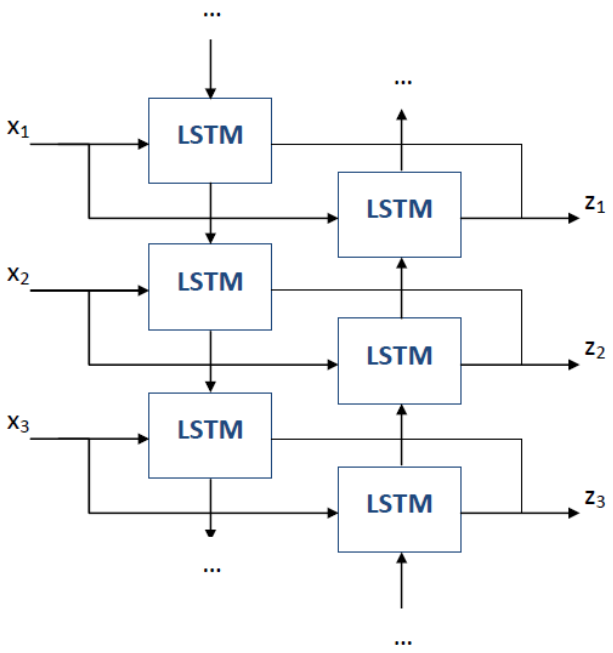


Figure 6.1: Scheme of the architecture of the Bi-LSTM network.

To be precise, we opted for a Joint Attentive Bidirectional Long Short-Term Memory (AB-LSTM) network (Sahu and Anand, 2018). This was employed by Sahu and Anand (2018) for a Drug-Drug Interaction (DDI) extraction task where, instead of finding drug-disease pairs, they found drug-drug pairs. In this architecture two Bi-LSTMs are trained: one with max pooling and the other with attentive pooling. Generally, pooling is used together with CNNs to find salient features regarding the class that can appear in different places (Goldberg and Hirst, 2017). In this case, it is used together with Bi-LSTM networks in order to obtain feature vectors of the same length (taking the last token output of the LSTM for can decrease the performance of the model on longer sequences). Finally, the resulting features are concatenated. The aim is to exploit the attention mechanism to capture important clues. The layers of this Joint AB-LSTM are depicted in Figure 6.2 and each of them is explained below.

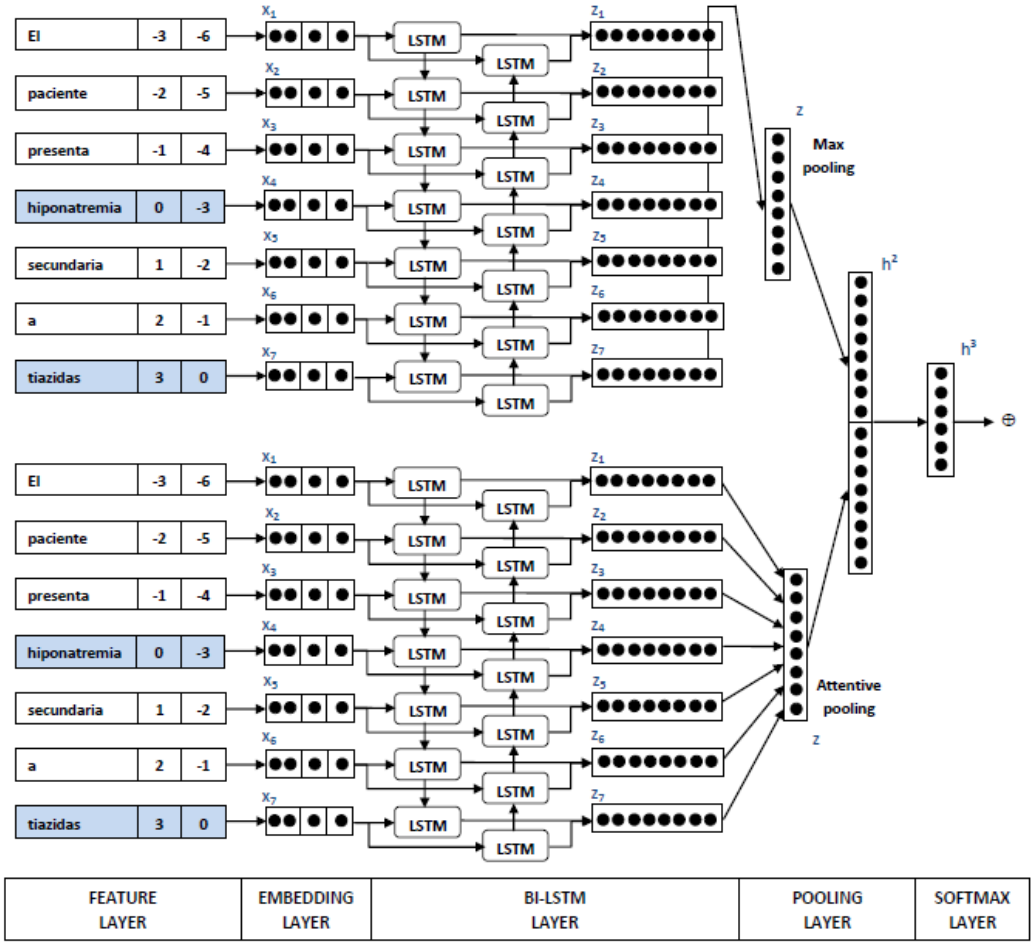


Figure 6.2: Scheme of the Joint AB-LSTM employed for the ADR detection. The features related with the entities are highlighted in light blue.

1. **Feature Layer**: This layer is designed to extract a set of core-features relevant to predicting ADR relations. The input of the system is the raw text, sentence by sentence. Then, for each word the following features are considered: i) Word (w): the corresponding word-form, ii) Distance_1 (d1): the number of words from the disease to the current word, iii) Distance_2 (d2): the number of words from the drug to the current word. Note that Sahu and Anand (2018) observed that the distances were relevant in the representation.

2. **Embedding Layer**: The symbolic features extracted from the previous layer (w, d1, d2) are transformed into dense ones. To do this, the corresponding embedding is retrieved for each feature and, next, all of them are concatenated, as is represented in expression (6.1). Here, x_t is the vector used to represent the t^{th} word of the sentence and comprises the embeddings of each symbolic feature mentioned above, denoted as $v1_t$, $v2_t$ and $v3_t$.

$$x_t = v1_t, v2_t, v3_t \tag{6.1}$$

3. **Bi-LSTM Layer**: The sequence of input vectors (x_t) is used to feed the non-linear classifier presented next, an LSTM, which is a type of RNN (Elman, 1990). The RNN is able to process sequential data and contains hidden states that store information about the previous states. However, the RNN is hard to train effectively because of the vanishing gradients problem (Pascanu et al., 2013). That is, the gradient of the optimization techniques (errors) tend to be smaller as the back-propagation process happens and does not learn temporally distant events. As a consequence, the RNN hardly captures long-range dependencies. The LSTM (Hochreiter and Schmidhuber, 1997) architecture was designed to solve the vanishing gradients problem. The main idea is to introduce gating mechanisms that can preserve gradients across time, the information is added to the memory cells and the gates regulate the information through them. Nevertheless, the LSTM only considers information about the previous states. The Bi-LSTM, a type of Bi-RNN (Schuster and Paliwal, 1997), allows to encompass information about the forward and backward states generated by two LSTMs, considering information about the past and the future. The LSTM is described formally in (6.2), where i_t, f_t, o_t are the input, forget and output gates respectively (being t the t^{th} word of the sentence). c_t is the current memory cell state, h_t is the current hidden state, c_{t-1} is the previous memory cell state and h_{t-1} is the previous hidden state. U_i, U_f, U_o, U_c are weight matrices of the recurrent connections, W_i, W_f, W_o, W_c are weight matrices of the input connections and b_i, b_f, b_o, b_c are bias vectors associated with corresponding gates and states (being i for the input gate, f for the forget gate, o for the output gate and c for the memory cell). σ is the sigmoid activation function, tanh is the

hyperbolic tangent function and $*$ is the element-wise product.

$$\begin{aligned} i_t =& \sigma(U_i x_t + W_i h_{t-1} + b_i) \\ f_t =& \sigma(U_f x_t + W_f h_{t-1} + b_f) \\ o_t =& \sigma(U_o x_t + W_o h_{t-1} + b_o) \\ c_t =& c_{t-1} * f_t + i_t * \tanh(U_c x_t + W_c h_{t-1} + b_c) \\ h_t =& \tanh(c_t) * o_t \end{aligned} \tag{6.2}$$

For a Bi-LSTM, the final output for the t^{th} word (z_t) is the concatenation of the backward and forward LSTM as shown in (6.3), where h_t^l corresponds to the left (backward) LSTM and h_t^r corresponds to the right (forward) LSTM.

$$z_t = h_t^l, h_t^r \tag{6.3}$$

4. **Pooling Layer**: Pooling is used to capture the most relevant features reducing the dimensions of the feature vector and is generally employed with CNNs (Goldberg and Hirst, 2017). Different pooling strategies can be applied: i) attentive pooling, ii) max pooling, and iii) average pooling. Attentive pooling takes the optimal information based on a weighted linear combination of feature vectors, where the weights are assigned according to the importance of these features. This is represented in (6.4), where Z is the output matrix of the Bi-LSTM layer, w_a is the learning parameter and α is the attention weight vector. An important point is that α would be different for every sentence, indicating that relevant context words may appear in distinct positions in different sentences.

$$\begin{aligned} H =& \tanh(Z) \\ \alpha =& \text{softmax}(w^{aT} H) \\ z =& \alpha Z^T \end{aligned} \tag{6.4}$$

Max pooling obtains the position with the most important information across the entire sentence. This is represented in (6.5), where z_t is the output of the Bi-LSTM layer for the t^{th} word and m is the total number of words in the sentence.

$$z = \max_{1 \leq t \leq m} [z_t] \tag{6.5}$$

Average pooling obtains the optimal by computing the average of feature vectors. This is represented in (6.6), where where z_t is the output of the Bi-LSTM layer for the t^{th} word and m is the total number of words in the sentence.

$$z = \frac{1}{m}\sum_{t=1}^{m}[z_t] \tag{6.6}$$

This implementation used two pooling approaches (max pooling and attentive pooling). Each pooling technique is applied to a Bi-LSTM and then, the features obtained with each one are concatenated. We explored each pooling approach separately and their combinations.

5. **Softmax Layer**: The tanh activation function is applied to the output of the pooling layer to obtain the input of the fully connected layer. There are other activation functions such as sigmoid or ReLU. In general, both ReLU and tanh significantly outperform the sigmoid activation function (Goldberg and Hirst, 2017), but ReLU is commonly used with CNNs (Akhtyamova et al., 2017). The Softmax function was applied in the output of the fully connected layer for prediction, that is, to obtain the probability distribution over the possible classes and select the class with maximum probability. This is shown in (6.7), where h^2 is the output of the pooling layer, h^3 is the input of the fully connected layer, W and b are the weight matrix and bias vector, respectively and $\hat{y}$ is the class that maximizes the probability.

$$\begin{aligned} h^3 &= \tanh(h^2) \\ p(y|x) &= \text{softmax}(Wh^{3T} + b) \\ \hat{y} &= \underset{y\in\{\oplus,\ominus\}}{\text{argmax}}\, p(y|x) \end{aligned} \tag{6.7}$$

So far, we explained the layers of the Joint AB-LSTM network. Regarding the training, the cross-entropy loss function was optimized by means of the Adam algorithm (Kingma and Ba, 2015). The Adam algorithm was designed for gradient-based optimization of stochastic objective functions and is an adaptive learning rate method, that is, it computes individual learning rates for different parameters. The Adam algorithm shows fast convergence while maintaining robustness in the choice of the learning rate (Goldberg and Hirst, 2017). We employed two regularization methods: i) L2-regularization, which places a squared penalty on parameters with large values by adding

an additive term to the loss function and ii) Dropout (Hinton et al., 2012), which randomly drops a set of neurons (features) in each training case. In general, the regularization methods are used to reduce the over-fitting that can occur in the neural networks (Goldberg and Hirst, 2017). We also explored the use of **Batch Normalization**. It is used to reduce the internal covariate shift, the change in the distribution of network activations due to the change in network parameters during training (Ioffe and Szegedy, 2015). This mechanism allows the use of higher learning rates, reduces the need for dropout and can also speed up the training process.

In addition, we observed that the **class imbalance** was tackled in some of the aforementioned related works. Stanovsky et al. (2017) applied Synthetic Minority Oversampling TEchnique (SMOTE), Lee et al. (2017) found an optimal threshold for prediction probabilities, Akhtyamova et al. (2017) assigned more weight on the output of the minority class, Luo et al. (2017) applied sub-sample and Masino et al. (2018) formed each batch by randomly selecting half of the batch from the positive instances and half of the batch from the negative instances. In our case, we explored the following techniques: i) re-sample, which obtains the same number of instances for the positive and the negative class by over-sampling the minority class and sub-sampling the majority class, ii) re-sample per batch, which is the aforementioned technique but applied to the instances of each batch, and iii) cost-sensitive learning, which assigns weights, inversely proportional to each class distribution, to the outputs of the network.

6.3 Results

In this section we present the results obtained in the experiments carried out with the Joint AB-LSTM. The hyper-parameters of the Joint AB-LSTM were fine-tuned. The values of the learning rate, dropout and L2-regularization were selected by means of a grid-search using batches of size 100 and hidden layers of size 200 and we explored a maximum of 18 epochs applying early-stopping criteria (the values by default in the software of Sahu and Anand (2018)). For the learning rate and L2-regularization the values explored ranged from 0 to 1 with geometric increments (0.0001, 0.001, 0.01, 0.1) and for the dropout from 0 to 1 with 5 steps of constant length equal to 0.2 (0.2, 0.4, 0.6, 0.8). Having selected these values, we also explored different hidden layers sizes with a grid-search from 50 to 300 with steps of 50 (100, 150,

200, 250) and different epochs and batches sizes with other grid-search using the values 18, 38 and 100 for the epochs and 50 and 100 for the batch sizes (for the number of epochs it is generally used a value around 100, but in some works lower values were used such as 40 (Jagannatha and Yu, 2016a), 30 (Gupta et al., 2018) or even 18 (Cocos et al., 2017)). The deep learning approaches were implemented using the TensorFlow package (Abadi et al., 2015) in Python.

Turning to practical details, we used pre-trained word-embeddings to represent the words and for the distances the embeddings were initialized with random values. These were created on the basis of the unannotated dataset formed by EHRs, with 109,618,393 word-forms and a vocabulary of 286,984 words (see Section 3.2.2). We made use of GloVe (Pennington et al., 2014) with a window of size (s) 10 and yielding vectors of 300 dimensions. In case of finding OOV words during the initialization, these values were initialized randomly. Regarding the preprocessing, the digits were replaced with 'DG' and the words were changed to lowercase. The short sentences were extended with padding 'PAD' until reaching the maximum length. The lemmatized versions of the corpora were created using FreeLing-Med (Oronoz et al., 2013). To assess the models we used the IxaMed-GS corpus and the hold-out evaluation scheme (see Section 3.3.1). Note that the training process of the Joint AB-LSTM model entails some randomness, for example, in the batch selection. Thus, the results obtained in several runs can differ slightly. For this reason, we decided to make three runs and provide the averaged results.

First, we created a **baseline** with a Feed Forward Neural Network (FFNN) using the embeddings generated with word-form as core-features. A FFNN might seem the natural architecture to cope with dense features in the context of ADR detection. Specifically, this FFNN could be seen as a simplified version of the Joint AB-LSTM that does not make use of the Bi-LSTM layer, depicted in Figure 6.2. Thus, the ability to cope with the context is diminished. The results of the baseline for the IxaMed-GS corpus are given in Table 6.1, where we can see that the f-measure achieved for the positive class was 36.5. Surprisingly, despite employing a neural network, the results of this FFNN underperformed from 46.2 to 36.5 those obtained in Chapter 4 with the best performing model inferred with RF using the symbolic representation (see Table 4.5) and underperformed from 63.9 to 36.5 those obtained in Chapter 5 with the best performing model inferred with RF using dense representation (see Table 5.4). Furthermore, we can observe that this approach

is unstable. For example, the standard deviation for the f-measure of the positive class is quite high, 8.2.

Approach	Features	Precision	Recall	F-measure	Class
FFNN	word-forms	89.3±11.1	23.3±6.7	36.5±8.2	⊕
		85.4±1.1	99.3±0.8	91.8±0.6	⊖
		86.1±1.9	85.4±1.1	81.7±1.9	W. Avg.
		85.4±1.1	85.4±1.1	85.4±1.1	Micro Avg.
		87.3±5.4	61.2±3.3	64.1±4.4	Macro Avg.

Table 6.1: Baseline results (mean and standard deviation) obtained for the dev set of the IxaMed-GS corpus with the FFNN.

Next, we developed the experiments with the Joint AB-LSTM network. We assessed different core-features (embeddings generated with **word-forms** and **lemmas**) and the impact of **Batch Normalization**. In addition, we assessed the techniques explored to tackle the **class imbalance**. The results achieved are shown in Table 6.2. The best performing approach employed lemmas and Batch Normalization, as well as cost-sensitive learning to tackle the class imbalance. This outperformed the most simple approach, increasing the f-measure of the positive class from 68.0 to 78.8. However, this approach is more unstable that the second best one, which does not overcome the class imbalance. The second best approach achieved an f-measure for the positive class of 76.3, but the standard deviation decreased from 3.4 to 1.3. Then, we decided to continue our experimentation without tackling the class imbalance.

Finally, we assessed different **pooling** strategies with the experiment that included embedded lemmas and Batch Normalization. Widely used pooling strategies in NLP include max, average and attentive pooling (Goldberg and Hirst, 2017). We explored the impact of each of them separately and also their combinations. The results given in Figure 6.3 show that the combination of both max and attentive pooling (the used in the previous experiments given in Table 6.2) provided the best performance, being max pooling the approach that offered a better performance separately.

Approach	Features	BN	Class imbalance	F-measure⊕
Joint AB-LSTM	word-forms	without	-	68.0±5.6
		with	-	74.5±2.2
			re-sample	60.4±2.0
			re-sample per batch	60.7±1.7
			cost-sensitive	73.2±4.5
	lemmas	without	-	74.1±1.6
		with	-	76.3±1.3
			re-sample	60.3±2.9
			re-sample per batch	63.9±3.7
			cost-sensitive	78.8±3.4

Table 6.2: F-measure of the positive class (mean and standard deviation) obtained for the dev set of the IxaMed-GS corpus using the Joint AB-LSTM network. Different features (word-forms and lemmas), the impact of Batch Normalization (denoted as "BN") and the approaches used to tackle the class imbalance (re-sample, re-sample per batch and cost-sensitive learning) are assessed.

In brief, the Joint AB-LSTM with embedded lemmas, Batch Normalization and the combination of max and attentive pooling achieved the best performance. Table 6.3 gives full details of the best performing model for the dev and test sets. For the dev set, the f-measure of the positive class is 76.3. For the test set, the f-measure of the positive class is 71.9. In both cases the results obtained in Chapter 5 (see Table 5.4) were improved. The f-measure of the positive class increased from 63.9 to 76.3 in the dev set and from 55.4 to 71.9 in the test set.

In addition, we also analyzed the ROC curve and the AUC. Figure 6.4 shows the ROC curves and the AUCs of the aforementioned experiments for the dev and test sets. Note that we calculated the mean of the results obtained in three evaluations. For this reason, there are three ROC curves in the same graphic. In both cases the points are above the diagonal and the averaged AUC outperforms the AUC obtained in Chapter 5 (see Figure 4.6), from 0.87 to 0.88 in the dev set and from 0.86 to 0.93 in the test set. Unexpectedly, the AUC of the test set is better than the AUC of the dev set, contrary to the happened with the f-measure.

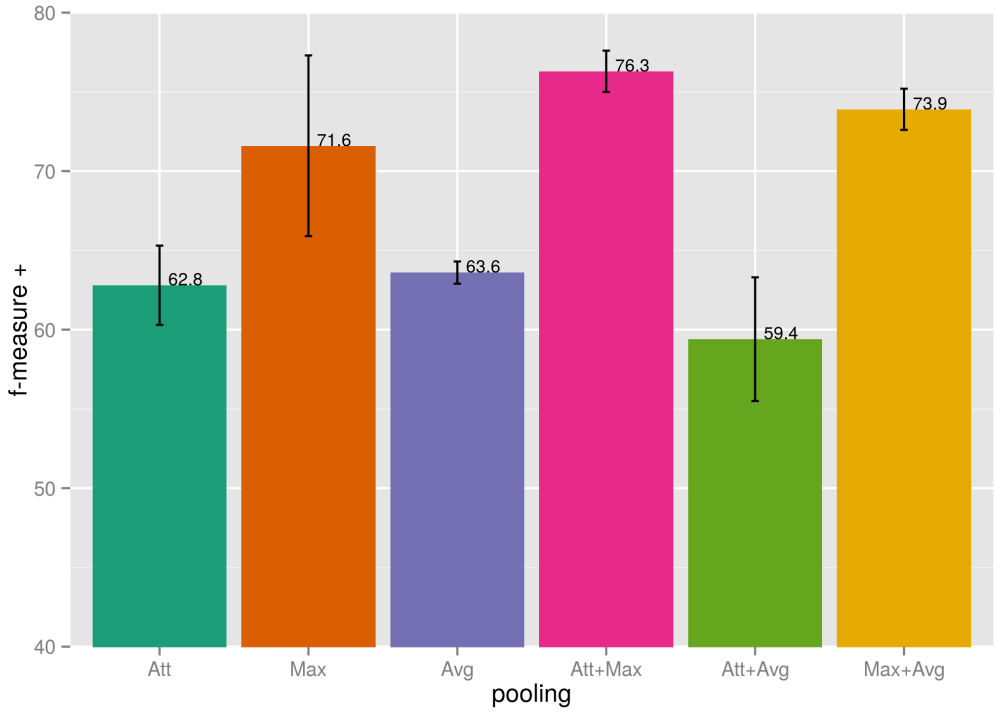


Figure 6.3: F-measure of the positive class obtained for the dev set of the IxaMed-GS corpus using the Joint AB-LSTM network. Different pooling strategies were assessed using the configuration that includes embedded lemmas and Batch Normalization. Attentive pooling is denoted as Att, Max pooling is denoted as Max and Average pooling is denoted as Avg. Note that the f-measure of the positive class is represented from 40.

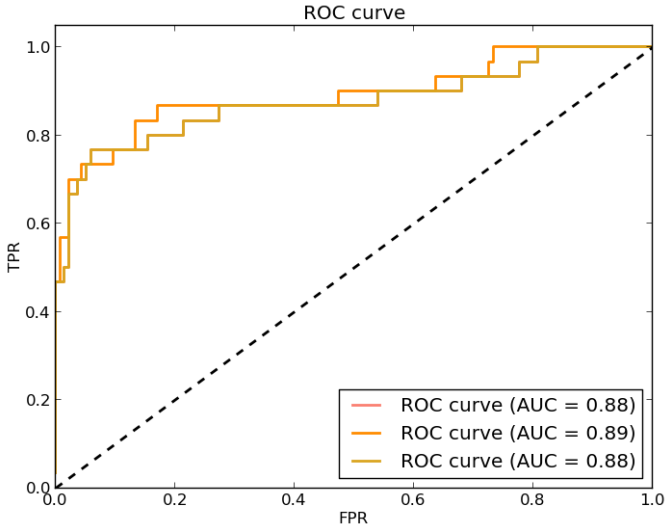
Precision	Recall	F-measure	Class
87.2±0.3	67.8±1.9	76.3±1.3	$\oplus$
93.2±0.3	97.8±0.0	95.4±0.2	$\ominus$
92.1±0.3	92.3±0.3	91.9±0.4	W. Avg.
92.3±0.3	92.3±0.3	92.3±0.3	Micro Avg.
90.2±0.3	82.8±1.0	85.8±0.8	Macro Avg.

(a) Model inferred with the train set and evaluated with the dev set.

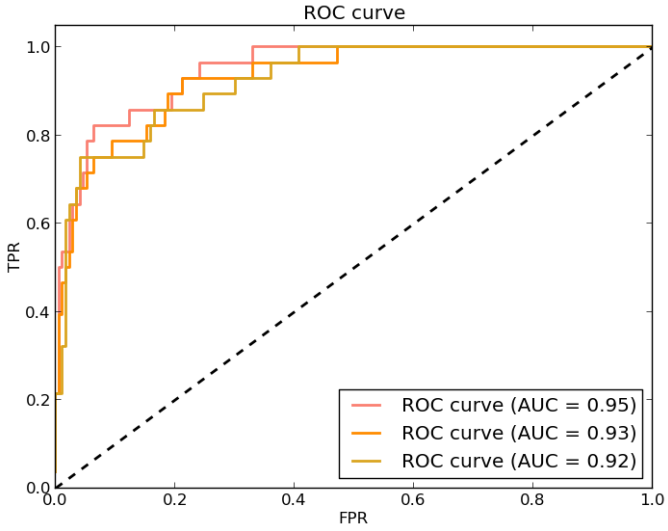
Precision	Recall	F-measure	Class
72.4±4.1	71.4±0.0	71.9±2.0	$\oplus$
95.3±0.1	95.5±0.9	95.4±0.5	$\ominus$
92.0±0.6	92.1±0.8	92.0±0.7	W. Avg.
92.1±0.8	91.8±1.0	92.7±0.8	Micro Avg.
83.8±2.1	83.4±0.4	83.6±1.2	Macro Avg.

(b) Model inferred with the train and dev sets and evaluated with the test set.

Table 6.3: Results (mean and standard deviation) of the best performing model inferred with the IxaMed-GS corpus and the Joint AB-LSTM network using lemmas and Batch Normalization.



(a) Model inferred with the train set and evaluated with the dev set.



(b) Model inferred with the train and dev sets and evaluated with the test set.

Figure 6.4: ROC curves and AUCs of the best experiment (embedded lemmas and Batch Normalization). The model was inferred with the IxaMed-GS corpus and the Joint AB-LSTM network. There are 3 ROC curves and AUCs because the evaluation was done with 3 runs.

6.3.1 Discussion

We used a **FFNN** as baseline, a simplified version of the Joint AB-LSTM that skips the Bi-LSTM layer. This was outperformed by the **Joint AB-LSTM**, which corroborates that the high-level features inferred by context-aware architectures (as is the case of the Bi-LSTM network) are important for the improvement of ADR detection. A drawback of the neural networks is that, although we did not have to resort to a complex manual feature engineering, we needed to explore a lot of hyper-parameters to adjust the neural network and obtain good results (Luo, 2017).

We found that **Batch Normalization** was helpful. In the previous experiments (see Table 6.2), the best performing value for the learning rate was 0.001 without Batch Normalization and 0.01 with Batch Normalization. This could confirm that the use of Batch Normalization allows higher learning rates (Ioffe and Szegedy, 2015). With regard to the optimal value for the dropout, this was 0.6 without Batch Normalization and 1.0 with Batch Normalization. This finding could also confirm that the need for dropout was reduced (Ioffe and Szegedy, 2015). Unfortunately, we also observed that the standard deviations of the results obtained in three runs were high.

A key issue in this work was to apply a mechanism to deal with high lexical variability. We assessed experimentally the use of **word-forms** and **lemmas** as core-features provided to the feature layer (see Table 6.2). We found that lemmatization was effective. It seemed that lemmas helped to overcome lexical variability, even though deep neural networks make use of high-level features inferred automatically. Note that the lemmas were a relevant features for the symbolic characterization (see Table 4.1).

Regarding the class imbalance, in the experiments carried out using the Joint AB-LSTM network without any mechanism to deal with the class imbalance, we obtained better results than with the best approach found with RF, that incorporated mechanisms against this issue. This could mean that the Joint AB-LSTM without external mechanisms remained robust against imbalanced classes.

In addition, we compared different **pooling strategies** such as max, average and attention pooling (Goldberg and Hirst, 2017) separately and also their combinations. According to the results, it seemed as if max and attention pooling complimented each other. This did not happen with the combination of attentive pooling and average pooling which offered worse results separately than max pooling. According to Suárez-Paniagua and

Segura-Bedmar (2018), that also analyzed the different pooling strategies, this can be due to the padding applied to the sentences, which can affect to the representation.

6.3.2 Error Analysis

After that, we inspected manually the predictions given by the best performing model, which made use of the Joint AB-LSMT with embedded lemmas and Batch Normalization.

With respect to the ADRs of the example given for the best performing model obtained with the symbolic representation in Chapter 4, all of them were also detected. In the sentence shown in Figure 6.5 we can see that the pairs '*hipoglucemia* - *septrin*' and '*hipoglucemia* - *timetropin*' were detected (the pairs '*hipoglucemia* - *novonorm*' was incorrectly detected as ADRs, one less than with the dense characterization). Note that the black arrows "Causada_por" correspond to the ADRs annotated by the experts and the red arrows "Causada_por_system" correspond to the predictions made by the system.

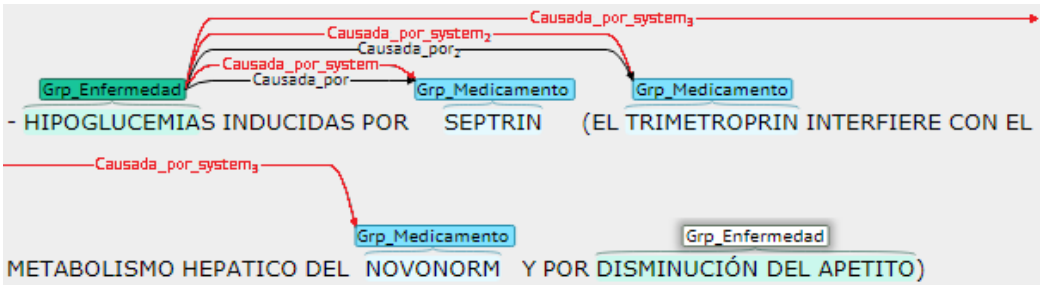


Figure 6.5: Example of sentence in which the model inferred with the Joint AB-LSTM and the embedded lemmas detected the ADRs discovered by the symbolic characterization. The sentence means 'Hypoglycemias induced by septrin (the trimethoprim interferes with the hepatic metabolism of the novonorm and by decreased appetite)'.

With respect to the ADR of the example given for the best performing model obtained with the smoothed dense representation in Chapter 5, it was also detected. In the sentence shown in Figure 6.6 we can see that the pair

'*episodio alérgico* - *contraste iodado*' was detected (the pair '*episodio alérgico* - *corticoides*' was incorrectly detected as ADR).

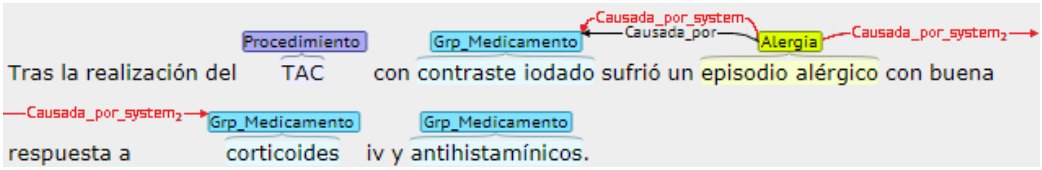


Figure 6.6: Example of sentence in which the model inferred with the Joint AB-LSTM and the embedded lemmas detected correctly the ADR discovered by the smoothed dense characterization. The sentence means 'After the realization of the CT scan with iodine contrast he suffered an allergic attack with a good response to iv corticoids and antihistamines.'.

Furthermore, we found that it was able to detect ADRs that were not found by the aforementioned models. For example, in the sentence shown in Figure 6.7 the ADR '*deterioro de la función renal* - *Septrin Forte*' was detected. This could corroborate that deep learning algorithms help to generalize, because this ADR was not seen during the training and was also detected the ADR '*insuficiencia renal* - *Septrin Forte*', as happened with the smoothed dense representation. Both diseases correspond to renal diseases.

We also observed several sources of errors. First, in **long sentences** (around 20 words) with a wide combination of potential events (around 5), we found FPs.

Second, some pairs were evaluated as FPs due to the errors made by the **experts** annotating the ADR relations. This confirms again that gold-standards are not necessarily free from errors (Perotte et al., 2014), the annotation is not trivial and can generate slight discrepancies among the experts. For example, in the sentence shown in Figure 6.8 the pair '*edema angioneurótico* - *iecas*' was not labeled as ADR by the experts when "edema angioneurótico" is the same disease that "angioedema", producing an FP (the ADR '*edema angioneurótico* - *iecas*' was detected correctly).

In addition, we continued finding FNs in speculative sentences with medical **uncertainty** (Velupillai and Kvist, 2012), that is, sentences were the diagnosis of the doctor about an ADR is uncertain.

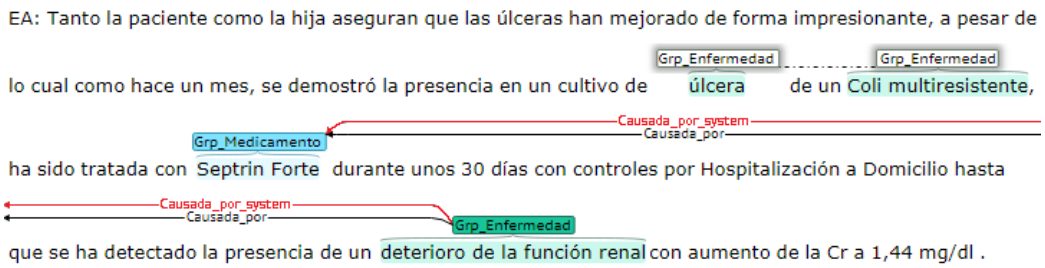


Figure 6.7: Example of sentence in which the model inferred with the Joint AB-LSTM and the embedded lemmas detected correctly the ADR annotated by the experts. The sentence means 'Current State: the patient as well as the daughter assure that the ulcers have improved in an impressive way, despite of this as a month ago, it was demonstrated the presence in a culture of ulcer of a multi-resistant Coli, she has been treated with Septrin Forte for 30 days with controls for Home Hospitalization until it has been detected the presence of a deterioration of kidney function with increase of the Cr to 1,44 mg/dl.'.

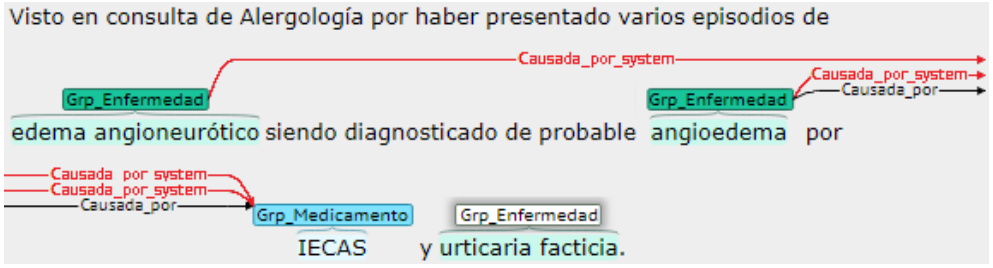


Figure 6.8: Example of sentence in which the best performing model committed an FP because the experts did not annotated some ADR relations. The sentence means 'Seen in Allergology surgery for having presented several episodes of angioneurotic edema being diagnosed of probable angioedema for ACE inhibitors and factitious urticaria.'.

In brief, with deep learning the system was able to detect more ADRs, which were not seen during the training. We found as sources of FPs long sentences with a wide combination of drug-disease pairs, but we did not observe FPs related with the word treatment. Furthermore, we observed that we can discover ADRs initially omitted by the experts. The sources of FNs were also speculative sentences.

6.4 Conclusions

6.4.1 Concluding remarks

In this chapter we employed neural networks to detect ADRs in real EHRs, to be precise a Joint AB-LSTM. The Joint AB-LSTM outperformed the results obtained with the RF classifier in Chapter 4 and Chapter 5. Our impression is that this happened because the Bi-LSTM architecture helped to represent information about the context in the inferred high-level features. The embedded lemmas were useful to improve the performance of the system by reducing the lexical variability. In addition, the Joint AB-LSTM seemed robust against the imbalanced distribution of the classes.

Taking into account these results, we answered to the following research questions:

Research Question 5

Can the Bi-LSTM networks help to cope with the lexical variability given that the dense features used to characterize the ADRs are inferred together with the model?

The FFNN does not outperform traditional classifiers such as Random Forest with smoothed dense features. However, the Bi-LSTM network outperforms them, particularly when embedded lemmas are used as core-features. It seems that the information captured from the context by the Bi-LSTM networks is relevant for the ADR detection and improves the generalization ability, which is helpful to cope with lexical variations.

Research Question 6

Are the Bi-LSTM networks sensitive to the class imbalance present in ADR detection?

With the traditional machine learning algorithms, the results are much lower if we do not tackle the class imbalance. By contrast, the approaches explored to tackle this problem in neural networks, such as re-sample or cost-sensitive learning, seem unnecessary and can even deteriorate the performance of the ADR detection model. In fact, Bi-LSTM networks can detect the majority of the ADRs present on the EHRs, without applying any mechanism to overcome the class imbalance.

Open questions. To develop the ADR detection, we employed the IxaMed-GS corpus to infer and evaluate all the models. The weak point of this study rests on the fact that the corpus does not have a high number of documents. Maybe, there are few examples to train and evaluate the model and we wonder whether this approach could be generalizable using more examples. Furthermore, the drug-disease pairs were created according to the gold mentions in order to focus only on the ADR detection. We know that the use of entities detected automatically leads to drop in performance of the ADR extraction. So far we explored an upper boundary of the system. For this reason, in the following chapter we intend to see the influence of larger corpus and automatically recognized entities in the performance of our ADR extraction system.

6.4.2 Publications

This work lead to the following publication:

1. Sara Santiso, Alicia Pérez, and Arantza Casillas. Exploring Joint AB-LSTM with embedded lemmas for Adverse Drug Reaction discovery. *IEEE Journal of Biomedical and Health Informatics*, 1–8, 2018.

7 Tolerance of Adverse Drug Reaction detection to noise

7.1 Introduction

In the previous chapters we explored three different strategies to detect ADRs as relations between the causative drug and the caused disease: i) symbolic representation with RF, ii) dense representation with RF, and iii) dense representation with Joint AB-LSTM. That is, for ADR detection we explored two types of representations, symbolic and dense, and two types of classifiers, traditional machine learning algorithms and deep learning algorithms.

First, the model was inferred using the Random Forest classifier and a symbolic representation (symbolic + RF). We created the symbolic representations of the intra-sentence and inter-sentence ADRs. Due to the skewed distribution of the class, we resorted to techniques to tackle the class imbalance. Despite this, we also had to reduce the imbalance by restricting to intra-sentence ADRs. At the end, the best performing model was obtained using intra-sentence relations and re-sample. This approach was explained in Chapter 4. Second, we created the model using the Random Forest classifier and a dense representation (dense + RF). We created the embeddings with different unannotated corpora and embedding generation approaches. These were used to represent the ADR candidates by means of concatenation and by means of context-aware embeddings. We also applied to these representations several smoothing techniques. At the end, the best performing model was obtained using context-aware embeddings created with GloVe, balanced

with re-sample, and applying truncation, director cosines and PCA. This approach was explained in Chapter 5. Third, we inferred the model with a dense representation and the Joint AB-LSTM classifier (dense + Joint AB-LSTM). We checked the influence of embedded lemmas and Batch Normalization. We also explored the robustness of the neural network architecture for the skewed class distribution. At the end, the best performing model was obtained using the embedded lemmas as core-features and Batch Normalization. This approach was explained in Chapter 6.

Along the experimentation, we observed how the performance of the ADR detection improved with each of the aforementioned strategies. That is, the best results were obtained with the Joint AB-LSTM and the dense representation. This is shown in Table 7.1, which gives the results obtained with the best performing model of each chapter. So far, we offered the results obtained with a hold-out evaluation in order to facilitate the reading. In this table we also offer the results obtained with the 10-fold cross-validation scheme in order to make possible to compare these results.

According to the results, the dense representation resulted useful since the f-measure of the positive class obtained with the symbolic representation improved in all the cases (from 46.2 to 63.9 in the dev set, from 43.2 to 55.4 in the test set and from 21.3 to 56.8 with 10-fold cross-validation). Furthermore, the abstract representation automatically inferred by the Joint AB-LSTM was even better, improving the dense representation in all the cases (from 63.9 to 76.3 in the dev set, from 55.4 to 71.9 in the test set and from 56.8 to 75.6 with 10-fold cross-validation).

We also developed significance tests with these experiments. In view of the related works (Botsis et al., 2011; Zhao et al., 2014, 2015; Henriksson et al., 2015b), we opted for the Wilcoxon test (Wilcoxon, 1945) and the Friedman test (Friedman, 1940). According to the Wilcoxon test, the differences between the results obtained in each fold of the 10-fold cross-validation were statistically significant with a confidence level of 99% in both comparisons: i) RF with symbolic and dense representations and ii) a dense representation with RF and Joint AB-LSTM. Furthermore, the differences among the three approaches were statistically significant with a confidence level of 99%, according to the Friedman test.

Approach		Precision	Recall	F-measure	Class
Features	Classifier				
Symbolic	RF	54.5	40.0	46.2	$\oplus$
		87.3	92.5	89.9	$\ominus$
		81.3	82.9	81.9	W. Avg.
		82.9	82.9	82.9	Micro Avg.
		70.9	66.3	68.0	Macro Avg.
Dense	RF	54.8	76.7	63.9	$\oplus$
		94.3	85.8	89.8	$\ominus$
		87.0	84.1	85.1	W. Avg.
		84.1	84.1	84.1	Micro Avg.
		74.5	81.2	76.9	Macro Avg.
Dense	Joint AB-LSTM	87.2±0.3	67.8±1.9	76.3±1.3	$\oplus$
		93.2±0.3	97.8±0.0	95.4±0.2	$\ominus$
		92.1±0.3	92.3±0.3	91.9±0.4	W. Avg.
		92.3±0.3	92.3±0.3	92.3±0.3	Micro Avg.
		90.2±0.3	82.8±1.0	85.8±0.8	Macro Avg.

(a) Hold-out: model inferred with the train set and evaluated with the dev set.

Approach		Precision	Recall	F-measure	Class
Features	Classifier				
Symbolic	RF	34.0	59.3	43.2	$\oplus$
		92.8	82.1	87.1	$\ominus$
		84.9	79.0	81.2	W. Avg.
		79.0	79.0	79.0	Micro Avg.
		63.4	70.7	65.2	Macro Avg.
Dense	RF	47.4	66.7	55.4	$\oplus$
		94.4	88.4	91.3	$\ominus$
		88.1	85.5	86.5	W. Avg.
		85.5	85.5	85.5	Micro Avg.
		70.9	77.6	73.4	Macro Avg.
Dense	Joint AB-LSTM	72.4±4.1	71.4±0.0	71.9±2.0	$\oplus$
		95.3±0.1	95.5±0.9	95.4±0.5	$\ominus$
		92.0±0.6	92.1±0.8	92.0±0.7	W. Avg.
		92.1±0.8	91.8±1.0	92.7±0.8	Micro Avg.
		83.8±2.1	83.4±0.4	83.6±1.2	Macro Avg.

(b) Hold-out: model inferred with the train and dev sets and evaluated with the test set.

Approach		Precision	Recall	F-measure	Class
Features	Classifier				
Symbolic	RF	18.3	25.5	21.3	⊕
		83.4	76.8	80.0	⊖
		72.4	68.1	70.0	W. Avg.
		68.1	68.1	68.1	Micro Avg.
		50.9	51.1	50.6	Macro Avg.
Dense	RF	54.6	59.1	56.8	⊕
		91.5	90.0	90.7	⊖
		85.2	84.7	85.0	W. Avg.
		84.7	84.7	84.7	Micro Avg.
		73.1	74.5	73.7	Macro Avg.
Dense	Joint AB-LSTM	81.3±1.3	71.7±3.7	75.6±2.8	⊕
		94.3±0.7	96.4±0.2	95.3±0.4	⊖
		92.0±0.8	92.1±0.7	91.9±0.8	W. Avg.
		92.1±0.7	92.1±0.7	92.1±0.7	Micro Avg.
		87.6±1.1	84.0±1.9	85.4±1.6	Macro Avg.

(c) 10-fold cross-validation: folds created with the train, dev and test sets.

Table 7.1: Results of each best performing approach (symbolic + RF, dense + RF, dense + Joint AB-LSTM) for the IxaMed-GS corpus.

The results given in Table 7.1 together with the significance tests corroborated that the Joint AB-LSTM with dense representation outperformed the rest of approaches. Nevertheless, we should also analyze how would change the performance of the ADR extraction under some variants. On the one hand, we are interested in discovering if the model is able to continue learning with larger corpora. As mentioned in Chapter 3, throughout this thesis more documents were acquired. However, they can be sightly different because some of them correspond to several hospitals with different specialties and were labeled by different experts. On the other hand, we feel curious about the deterioration of second step of the ADR extraction system (ADR detection) if we employ entities recognized automatically in the first step (MER). That is, using a real MER system instead of the gold mentions (the manual annotations given by the experts). So far we focused on the second step to get the performance of the ADR detection, but we are also interested on assessing the performance of the entire ADR extraction system as in real scenarios (turn to Figure 1.2 to see the two steps of the pipeline).

As a consequence, in this chapter we shall address the following research questions:

Research Question 7

How do the variations in the size and sub-domains of the corpus affect to the performance of the ADR detection model?

Research Question 8

How is the tolerance of the ADR detection model to the noise introduced by the automatic medical entity recognition?

The rest of the chapter is organized as follows: Section 7.2 gives detailed evaluations of the best performing experiment using corpora of different sizes and hospitals. Section 7.3 gives detailed evaluations of the best performing experiment with misrecognized drug-disease pairs due to the use a MER system. Section 7.4 provides the final conclusions.

7.2 Tolerance of ADR detection to corpus variations

According to the results shown in the previous section, we selected the Joint AB-LSTM as the best performing approach and our purpose was to assess the impact on it of slight variations in the data: i) different size of corpora (increase the number of EHRs), motivated by Akhtyamova et al. (2017) who mentioned that the size of the training corpora had some impact in the accuracy of a CNN model for ADR extraction, and ii) EHRs from different hospitals and annotated by different experts, motivated by Sarker and Gonzalez (2015) who observed that multi-corpus training can provide significant improvements in classification accuracies if the corpora used are compatible.

When we started to work on ADR detection, we just had the IxaMed-GS corpus and the number of documents was not high (although the number of words is comparable with other corpus (Pérez et al., 2017)). With this corpus we carried out the experimentation shown in Chapter 4, Chapter 5 and Chapter 6. During the development of this work, within the framework of the DETEAMI and PROSAMED projects, we got more EHRs leading to the IxaMed-CH corpus and the IxaMed-E corpus. In summary, IxaMed-GS (see Table 3.1) consists of 75 EHRs (41,633 words) from the Galdakao

hospital. This corpus has 110 positive instances and 538 negative instances. IxaMed-CH (see Table 3.2) consists of 267 EHRs (158,263 words) from the Galdakao and Basurto hospitals. This corpus has 338 positive instances and 3,087 negative instances. IxaMed-E (see Table 3.3) consists of 463 EHRs (230,040 words) from the Galdakao and Basurto hospitals. This corpus has 527 positive instances and 21,945 negative instances. Apart from increasing the number of instances in each one, curiously, the imbalance ratio (the ratio between positive and negative instances) is approximately three times higher from one to the other. To be precise, for IxaMed-GS is 1:4, for IxaMed-CH is 1:11 and for IxaMed-E is 1:33. Moreover, the are more OOV words when the corpus is larger. Turn to Section 3.2.1 for more details about the three corpora.

7.2.1 Results

With the aforementioned annotated corpora, we applied the Joint AB-LSTM classifier with a dense representation (using embedded lemmas and Batch Normalization). Given that in the experiments developed with the IxaMed-GS corpus in Chapter 6 it was not clear whether the use of cost-sensitive learning was beneficial to improve the performance of the Joint AB-LSTM, we decided to employ this mechanism to tackle the class imbalance with the three corpora.

Figure 7.1 shows the boxplots of these results. It is possible to see again that with the IxaMed-GS corpus the results were better using cost-sensitive learning than without tackling the class imbalance, but the dispersion was notably high. By contrast, with the IxaMed-CH and IxaMed-E corpora the results obtained with cost-sensitive learning were lower than without tackling the class imbalance. Furthermore, comparing the boxplots of the two best performing experiments, we can observe that the differences are not significant. Therefore, it resulted that the best Joint AB-LSTM implementation was the one that did not resort to any mechanism to overcome the class imbalance.

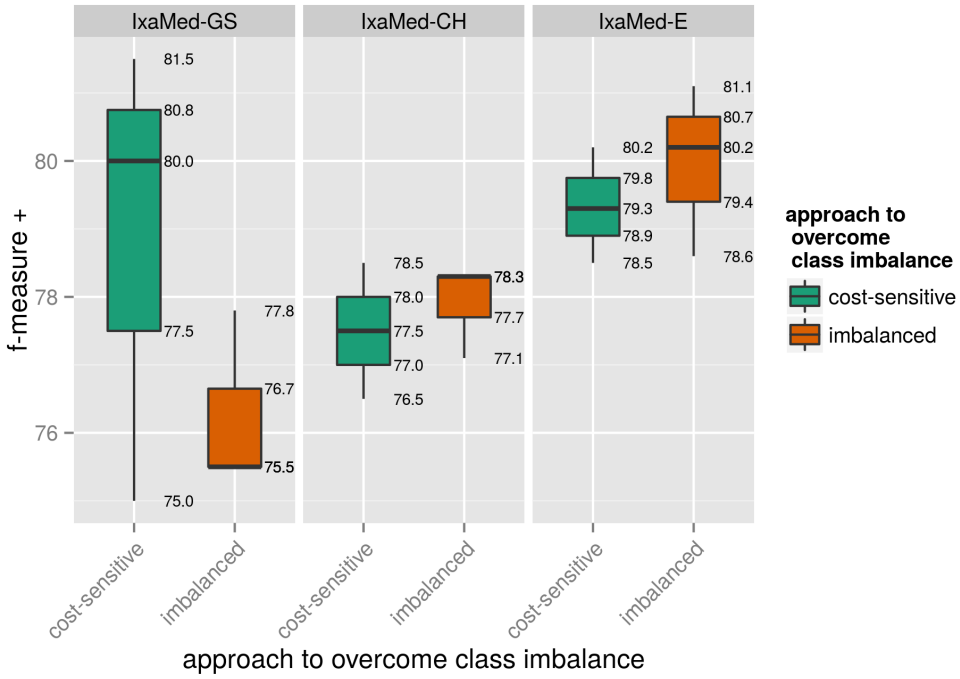


Figure 7.1: Boxplots of the f-measure for the positive class obtained in three runs of the Joint AB-LSTM (with embedded lemmas and Batch Normalization) for the dev set of each corpus (IxaMed-GS, IxaMed-CH, IxaMed-E). They show the results obtained using cost-sensitive learning (denoted as "cost-sensitive") and those obtained without applying any mechanism to overcome the class imbalance ("imbalanced").

Table 7.2 shows the detailed results obtained using these corpora (IxaMed-GS, IxaMed-CH and IxaMed-E) and the Joint AB-LSTM with embedded lemmas and Batch Normalization. In this case, the training of the models and the evaluation were done with the same corpus in each experiment. According to the f-measure of the positive class, the IxaMed-CH corpus outperformed the IxaMed-GS corpus from 76.3 to 77.9 in the dev set, from 71.9 to 73.3 in the test set and from 75.6 to 75.9 with 10-fold cross-validation. The IxaMed-E corpus outperformed the IxaMed-CH corpus from 77.9 to 79.9 in the dev set, from 73.3 to 75.2 in the test set and from 75.9 to 80.8 with 10-fold cross-validation. In brief, the models inferred with larger corpora outperformed those inferred with smaller corpora.

Corpus	Precision	Recall	F-measure	Class
IxaMed-GS	87.2±0.3	67.8±1.9	76.3±1.3	⊕
	93.2±0.3	97.8±0.0	95.4±0.2	⊖
	92.1±0.3	92.3±0.3	91.9±0.4	W. Avg.
	92.3±0.3	92.3±0.3	92.3±0.3	Micro Avg.
	90.2±0.3	82.8±1.0	85.8±0.8	Macro Avg.
IxaMed-CH	89.3±0.6	69.2±1.4	77.9±0.7	⊕
	94.1±0.2	98.3±0.2	96.2±0.1	⊖
	93.2±0.1	93.4±0.1	93.1±0.2	W. Avg.
	93.4±0.1	93.4±0.1	93.4±0.1	Micro Avg.
	91.6±0.2	83.7±0.6	87.0±0.3	Macro Avg.
IxaMed-E	90.3±0.8	71.8±2.3	79.9±1.3	⊕
	94.7±0.4	98.5±0.2	96.6±0.2	⊖
	94.0±0.3	94.2±0.3	93.9±0.3	W. Avg.
	94.2±0.3	94.2±0.3	94.2±0.3	Micro Avg.
	92.6±0.3	85.2±1.0	88.3±0.7	Macro Avg.

(a) Hold-out: model inferred with the train set and evaluated with the dev set.

Corpus	Precision	Recall	F-measure	Class
IxaMed-GS	72.4±4.1	71.4±0.0	71.9±2.0	⊕
	95.3±0.1	95.5±0.9	95.4±0.5	⊖
	92.0±0.6	92.1±0.8	92.0±0.7	W. Avg.
	92.1±0.8	91.8±1.0	92.7±0.8	Micro Avg.
	83.8±2.1	83.4±0.4	83.6±1.2	Macro Avg.
IxaMed-CH	76.0±2.8	70.9±5.6	73.3±2.2	⊕
	96.1±0.7	96.9±0.7	96.5±0.2	⊖
	93.7±0.5	93.8±0.3	93.7±0.4	W. Avg.
	93.8±0.3	93.8±0.3	93.8±0.3	Micro Avg.
	86.1±1.2	83.9±2.5	84.9±1.2	Macro Avg.
IxaMed-E	74.4±5.1	76.0±2.9	75.2±3.9	⊕
	96.5±0.5	96.2±0.9	96.3±0.6	⊖
	93.7±1.0	93.6±1.1	93.7±1.0	W. Avg.
	93.6±1.0	93.6±1.1	93.6±1.1	Micro Avg.
	85.4±2.7	86.1±1.9	85.8±2.3	Macro Avg.

(b) Hold-out: model inferred with the train and dev sets and evaluated with the test set.

Corpus	Precision	Recall	F-measure	Class
IxaMed-GS	81.3±1.3	71.7±3.7	75.6±2.8	⊕
	94.3±0.7	96.4±0.2	95.3±0.4	⊖
	92.0±0.8	92.1±0.7	91.9±0.8	W. Avg.
	92.1±0.7	92.1±0.7	92.1±0.7	Micro Avg.
	87.6±1.1	84.0±1.9	85.4±1.6	Macro Avg.
IxaMed-CH	83.2±0.7	70.2±0.5	75.9±0.2	⊕
	95.7±0.1	97.8±0.1	96.8±0.0	⊖
	94.1±0.0	94.3±0.0	94.1±0.0	W. Avg.
	94.3±0.0	94.3±0.0	94.3±0.0	Micro Avg.
	89.5±0.3	84.0±0.2	86.3±0.1	Macro Avg.
IxaMed-E	86.1±0.3	76.2±1.3	80.8±0.8	⊕
	95.2±0.2	97.4±0.0	96.3±0.1	⊖
	93.6±0.2	93.8±0.2	93.6±0.2	W. Avg.
	93.8±0.2	93.8±0.2	93.8±0.2	Micro Avg.
	90.6±0.2	86.8±0.6	88.5±0.5	Macro Avg.

(c) 10-fold cross-validation: folds created with the train, dev and test sets.

Table 7.2: Results of the best performing approach (dense + Joint AB-LSTM) with each corpus (IxaMed-GS, IxaMed-CH, IxaMed-E).

In addition, we made cross-corpus evaluations, that is to say, we inferred the model with the same corpus and assessed it with the three of them. This enabled us to compare the results obtained by each model with a fixed evaluation set. Table 7.3 shows these results. An inspection of this table shows that i) the same model tended to offer higher results when the evaluation set was smaller and lower results when the evaluation set was higher and ii) comparing each model under the same evaluation set, the results tended to improve when the model was inferred with larger corpora. For example, if we focus on the train set of IxaMed-E, the f-measure of the positive class for the dev set was 83.3 evaluating with IxaMed-GS, 82.2 with IxaMed-CH and 79.9 with IxaMed-E. If we focus on the evaluation made with the dev set of IxaMed-E, the f-measure of the positive class was 51.4 training with IxaMed-GS, 79.9 with IxaMed-CH and 79.9 with IxaMed-E.

Corpus		Precision	Recall	F-measure	Class
Training	Evaluation				
IxaMed-GS	IxaMed-GS	87.2±0.3	67.8±1.9	76.3±1.3	⊕
		93.2±0.3	97.8±0.0	95.4±0.2	⊖
		92.1±0.3	92.3±0.3	91.9±0.4	W. Avg.
		92.3±0.3	92.3±0.3	92.3±0.3	Micro Avg.
		90.2±0.3	82.8±1.0	85.8±0.8	Macro Avg.
	IxaMed-CH	79.0±3.3	53.0±3.8	63.3±2.9	⊕
		91.1±0.6	97.1±0.6	88.4±10.0	⊖
		89.1±0.8	89.7±0.7	88.9±0.8	W. Avg.
		89.7±0.7	89.7±0.7	89.7±0.7	Micro Avg.
		85.0±1.7	75.0±1.8	78.7±3.3	Macro Avg.
	IxaMed-E	54.7±3.5	41.0±3.1	51.4±8.2	⊕
		90.0±1.8	94.3±0.3	90.8±1.1	⊖
		84.2±1.0	85.6±0.6	84.4±0.5	W. Avg.
		85.6±0.6	85.6±0.6	85.6±0.6	Micro Avg.
		72.3±0.9	67.7±1.6	71.1±3.6	Macro Avg.
IxaMed-CH	IxaMed-GS	98.5±2.6	72.2±1.9	83.3±2.2	⊕
		94.2±0.4	99.8±0.4	96.9±0.4	⊖
		94.9±0.8	94.8±0.8	94.4±0.7	W. Avg.
		94.8±0.8	94.8±0.8	94.8±0.8	Micro Avg.
		96.3±1.5	86.0±1.2	90.1±1.3	Macro Avg.
	IxaMed-CH	89.3±0.6	69.2±1.4	77.9±0.7	⊕
		94.1±0.2	98.3±0.2	96.2±0.1	⊖
		93.2±0.1	93.4±0.1	93.1±0.2	W. Avg.
		93.4±0.1	93.4±0.1	93.4±0.1	Micro Avg.
		91.6±0.2	83.7±0.6	87.0±0.3	Macro Avg.
	IxaMed-E	95.7±1.6	68.7±1.8	79.9±0.6	⊕
		94.2±0.3	99.4±0.3	96.8±0.1	⊖
		94.5±0.1	94.4±0.1	90.7±5.7	W. Avg.
		94.4±0.1	94.4±0.1	94.4±0.1	Micro Avg.
		95.0±0.7	84.0±0.7	88.3±0.4	Macro Avg.
IxaMed-E	IxaMed-GS	98.6±2.5	72.2±1.9	83.3±1.1	⊕
		94.2±0.3	99.8±0.4	96.9±0.2	⊖
		95.0±0.4	94.7±0.4	94.4±0.3	W. Avg.
		94.7±0.4	94.7±0.4	94.7±0.4	Micro Avg.
		96.4±1.2	86.0±0.9	90.1±0.7	Macro Avg.
	IxaMed-CH	90.5±3.2	75.0±2.2	82.0±0.8	⊕
		95.1±0.4	98.4±0.7	96.7±0.2	⊖
		94.4±0.3	94.5±0.4	94.3±0.3	W. Avg.
		94.5±0.4	94.5±0.4	94.5±0.4	Micro Avg.
		92.8±1.5	86.7±0.9	89.3±0.5	Macro Avg.
	IxaMed-E	90.3±0.8	71.8±2.3	79.9±1.3	⊕
		94.7±0.4	98.5±0.2	96.6±0.2	⊖
		94.0±0.3	94.2±0.3	93.9±0.3	W. Avg.
		94.2±0.3	94.2±0.3	94.2±0.3	Micro Avg.
		92.6±0.3	85.2±1.0	88.3±0.7	Macro Avg.

(a) Hold-out: model inferred with the train set and evaluated with the dev set.

Corpus		Precision	Recall	F-measure	Class
Training	Evaluation				
IxaMed-GS	IxaMed-GS	72.4±4.1	71.4±0.0	71.9±2.0	⊕
		95.3±0.1	95.5±0.9	95.4±0.5	⊖
		92.0±0.6	92.1±0.8	92.0±0.7	W. Avg.
		92.1±0.8	91.8±1.0	92.7±0.8	Micro Avg.
		83.8±2.1	83.4±0.4	83.6±1.2	Macro Avg.
	IxaMed-CH	41.8±6.1	58.6±5.2	48.6±4.6	⊕
		94.0±0.7	88.7±2.6	91.3±1.5	⊖
		87.8±1.2	85.1±2.3	86.2±1.8	W. Avg.
		85.1±2.3	85.1±2.3	85.1±2.3	Micro Avg.
		67.9±3.2	73.6±2.7	69.9±3.0	Macro Avg.
	IxaMed-E	35.4±2.0	58.5±4.4	44.1±2.8	⊕
		93.4±0.7	84.5±0.4	88.7±0.4	⊖
		86.0±0.8	81.3±0.8	83.1±0.7	W. Avg.
		81.3±0.8	81.3±0.8	81.3±0.8	Micro Avg.
		64.4±1.4	71.5±2.3	66.5±1.6	Macro Avg.
IxaMed-CH	IxaMed-GS	84.5±2.2	71.4±3.6	77.4±2.5	⊕
		95.4±0.6	97.8±0.3	96.6±0.3	⊖
		93.8±0.6	94.1±0.6	93.9±0.7	W. Avg.
		94.1±0.6	94.1±0.6	94.1±0.6	Micro Avg.
		90.0±1.3	84.6±1.8	87.0±1.4	Macro Avg.
	IxaMed-CH	76.0±2.8	70.9±5.6	73.3±2.2	⊕
		96.1±0.7	96.9±0.7	96.5±0.2	⊖
		93.7±0.5	93.8±0.3	93.7±0.4	W. Avg.
		93.8±0.3	93.8±0.3	93.8±0.3	Micro Avg.
		86.1±1.2	83.9±2.5	84.9±1.2	Macro Avg.
	IxaMed-E	81.8±1.4	74.8±1.3	78.1±0.6	⊕
		96.4±0.2	97.6±0.3	97.0±0.1	⊖
		94.5±0.2	94.7±0.2	94.6±0.2	W. Avg.
		94.7±0.2	94.7±0.2	94.7±0.2	Micro Avg.
		89.1±0.6	86.2±0.6	87.6±0.3	Macro Avg.
IxaMed-E	IxaMed-GS	84.5±4.9	69.1±5.4	75.8±1.8	⊕
		95.0±0.8	97.8±0.9	96.4±0.2	⊖
		93.5±0.4	93.7±0.3	93.5±0.4	W. Avg.
		93.7±0.3	93.7±0.3	93.7±0.3	Micro Avg.
		89.7±2.1	83.4±2.3	86.1±0.9	Macro Avg.
	IxaMed-CH	81.6±3.4	82.8±2.5	82.1±0.6	⊕
		97.7±0.3	97.4±0.7	97.6±0.2	⊖
		95.8±0.2	95.7±0.3	95.7±0.2	W. Avg.
		95.7±0.3	95.7±0.3	95.7±0.3	Micro Avg.
		89.7±1.6	90.1±0.9	89.8±0.4	Macro Avg.
	IxaMed-E	74.4±5.1	76.0±2.9	75.2±3.9	⊕
		96.5±0.5	96.2±0.9	96.3±0.6	⊖
		93.7±1.0	93.6±1.1	93.7±1.0	W. Avg.
		93.6±1.0	93.6±1.1	93.6±1.1	Micro Avg.
		85.4±2.7	86.1±1.9	85.8±2.3	Macro Avg.

(b) Hold-out: model inferred with the train and dev sets and evaluated with the test set.

Table 7.3: Results of the best performing approach (dense + Joint AB-LSTM) with cross-corpus experiments. The models inferred with each corpus are assessed with the evaluation set of each corpus (IxaMed-GS, IxaMed-CH, IxaMed-E).

7.2.2 Discussion

In order to examine the results where the training and the evaluation of the experiments were done with the same corpus (see Table 7.2), we would like to remind that the **class imbalance** triples from one corpus to the other. Despite of the increasing skew, the results improved as the size of the corpus increased. Then, we could state that the Joint AB-LSTM was much more robust against unequal distributions than the traditional classifier used in the previous two approaches (Chapter 4 and Chapter 5).

While both IxaMed-CH and IxaMed-E preserve the domain, the **subdomains** are not exactly the same as in IxaMed-GS. Besides, the annotations might differ slightly as different experts were involved in the annotation process. Interestingly, the results suggested that the inference process was not affected and, as the corpora increased, the model was enhanced.

We also should bear in mind that the **lexical variability** is higher as the size of the corpus increases (see the OOVs of the three annotated corpora in Section 3.2.1). Nevertheless, the performance of the Joint AB-LSTM did not drop as the variability increases. That is, this approach seemed to manage with changes in the lexicon, at least, provided that the number of data increases.

The results of the cross-corpus evaluation shown in Table 7.3 would corroborate the previous analysis. That is, the generalization ability increased when the model was inferred using corpora with more instances.

7.2.3 Error analysis

Inspecting the predictions made by these models, we observed that sometimes a drug-disease pair detected as non-ADR by the model inferred with the IxaMed-GS corpus, was correctly detected as ADR by the model inferred with the IxaMed-CH corpus, avoiding FNs. For example, in the sentence given in Figure 7.2 the ADR '*hiperglucemia leve - tto. corticoideo*' is detected with IxaMed-CH although it was not detected with IxaMed-GS. Interestingly, this ADR was detected in the same way that the ADR '*diabete mellitus tipo 2 - tratamiento corticoideo*' was detected previously, despite of the fact that during the training the observed ADRs were '*diabetes mellitus - corticoterapia*', '*hiperglucemia - corticoides*', '*hiperglucemia - tratamiento corticoideo*'. That is, different terms were used to make reference to the same ADR. Note that the black arrows "Causada_por" correspond to the ADRs annotated by

the experts and the red arrows "Causada_por_system" correspond to the predictions made by the system.

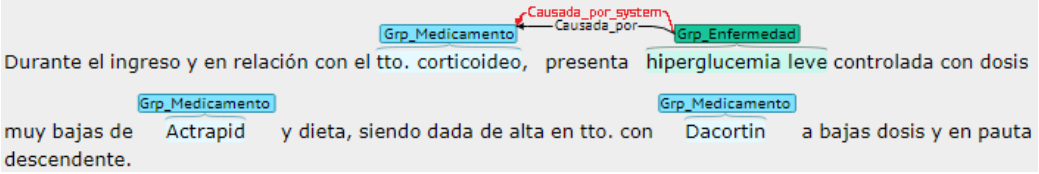


Figure 7.2: Example of sentence in which the ADR is detected with the IxaMed-CH corpus and it was not detected with the IxaMed-GS corpus. The sentence means 'During admission and in relation with the corticoidal treatment, he presents slight hyperglycemia controlled with very low doses of Actrapid and diet, being discharged in treatment with Darcotin at low doses and in downward trend.'.

We also observed that with the IxaMed-E corpus it was possible to detect ADRs such as '*cefalea* - *nitroglicerina*', which is shown in the sentence given in Figure 7.3.

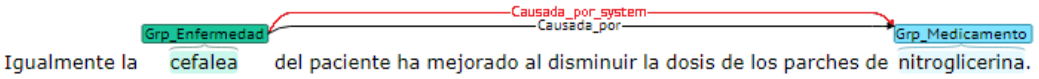


Figure 7.3: Example of sentence in which the ADR is detected with the IxaMed-E corpus. The sentence means 'Equally the headache of the patient has improved as the doses of the nitroglycerin patches decreased.'.

7.3 Tolerance of ADR detection to noise derived from MER

Our system is implemented as a pipeline with two steps (see Figure 1.2). First, the medical entities (e.g. drugs and diseases) are recognized. Second, the drug-disease pairs are classified as ADR or non-ADR. As a consequence, the errors generated in the MER step are propagated to the ADR detection step. With the aim of assessing the tolerance to noise of the ADR detection, we selected the Joint AB-LSTM with the IxaMed-E as the best performing

approach. First, we computed the performance of the ADR detection system with an ideal MER system, that is, using the gold mentions (the entities annotated by the experts). This enables to focus on the performance of the ADR detection as an isolated stage in an ideal situation. Next, we made use of a real MER system in the first step and assessed the performance of the ADR detection model. Finally, we introduced, gradually, random noise to the gold mentions in an automatic way. In this way, we compared the real MER with the noise that other MER systems could introduce.

In this case, to recognize automatically the entities we resorted to the CRF classifier (Lafferty et al., 2001), which is an algorithm for sequence labeling widely used for NER. There are more sophisticated approaches based on deep learning that reached promising results for entity recognition (Lample et al., 2016; Habibi et al., 2017; Ju et al., 2018; Weegar et al., 2018). Nevertheless, since our purpose was to get errors in order to assess their propagation and given that the MER itself is not the main objective of this work, we decided not to resort to the best MER approaches.

7.3.1 Results

To analyze the influence of the errors propagated from the automatic entity recognition on ADR detection, we carried out three experiments:

1. Using for MER an ideal system able to guess the mentions as an expert would do and a real system for automatic ADR detection (this is the situation explored so far in Table 7.2). This scenario allows us to assess the scope of the ADR detection.

2. Using a real system for automatic MER and an ideal system for ADR detection. In this way, we can derive the upper threshold achievable by a real ADR detection system with a real MER.

3. Using both, MER and ADR detection, real systems.

First, we assessed the MER system. The CRF classifier was created using the freely available implementation CRF++ (Kudo, 2005) and using as features the lemma, the POS and the semantic tag. The results obtained training with the train set and evaluating with the dev set (train vs dev) and the results obtained training with the train and dev sets and evaluating with the test set (train∪dev vs test) are shown in Table 7.4. We observed that f-measure was 57.4 and 57.2 respectively with exact-match. Turn to Appendix B for more information about the MER experiments.

	Exact			Partial		
	P	R	F	P	R	F
train vs dev	64.4	51.7	57.4	90.7	79.8	84.9
train∪dev vs test	63.6	52.1	57.2	89.6	80.4	84.8

Table 7.4: Precision (P), Recall (R) and F-measure (F) for MER using the CRF classifier with the IxaMed-E corpus.

After that, we obtained the results of the aforementioned three scenarios, which are shown in Table 7.5. Note that we employed the MER entities only in the evaluation sets (dev and test), as in a real system. As it was expected, the performance of the Joint AB-LSTM decreased from the first scenario to the third one, since we replaced the ideal MER system with the CRF (the f-measure of the positive class decreased from 79.9 to 74.5 in the dev set and from 75.2 to 65.7 in the test set). In fact, the second scenario, with CRF as MER and ideal ADR detection system, gives us the upper threshold performance in case of dropping entities (the f-measure of the positive class was 87.0 in the dev set and 77.1 in the test set). Note that the recall of the positive class in the second scenario was 76.9 in the dev set and 62.8 in the test set and not 100.0%. This happened because it was not possible to find all the ADR relations due to unlabeled entities.

ADR extraction		Precision	Recall	F-measure	Class
MER	ADR detection				
Ideal	Joint AB-LSTM	90.3±0.8	71.8±2.3	79.9±1.3	⊕
		94.7±0.4	98.5±0.2	96.6±0.2	⊖
		94.0±0.3	94.2±0.3	93.9±0.3	W. Avg.
		94.2±0.3	94.2±0.3	94.2±0.3	Micro Avg.
		92.6±0.3	85.2±1.0	88.3±0.7	Macro Avg.
CRF	Ideal	100.0±0.0	76.9±0.0	87.0±0.0	⊕
		95.7±0.0	100.0±0.0	97.8±0.0	⊖
		96.4±0.0	96.3±0.0	96.1±0.0	W. Avg.
		96.3±0.0	96.3±0.0	96.3±0.0	Micro Avg.
		97.9±0.0	88.5±0.0	92.4±0.0	Macro Avg.
CRF	Joint AB-LSTM	96.4±2.0	60.7±0.0	74.5±0.6	⊕
		92.9±0.0	99.5±0.3	96.1±0.1	⊖
		93.5±0.3	93.3±0.2	92.6±0.2	W. Avg.
		93.3±0.2	93.3±0.2	93.3±0.2	Micro Avg.
		94.7±1.0	80.1±0.2	85.3±0.4	Macro Avg.

(a) Hold-out: model inferred with the train set and evaluated with the dev set.

ADR extraction		Precision	Recall	F-measure	Class
MER	ADR detection				
Ideal	Joint AB-LSTM	74.4±5.1	76.0±2.9	75.2±3.9	⊕
		96.5±0.5	96.2±0.9	96.3±0.6	⊖
		93.7±1.0	93.6±1.1	93.7±1.0	W. Avg.
		93.6±1.0	93.6±1.1	93.6±1.1	Micro Avg.
		85.4±2.7	86.1±1.9	85.8±2.3	Macro Avg.
CRF	Ideal	100.0±0.0	62.8±0.0	77.1±0.0	⊕
		94.9±0.0	100.0±0.0	97.4±0.0	⊖
		95.5±0.0	95.3±0.0	94.8±0.0	W. Avg.
		95.3±0.0	95.3±0.0	95.3±0.0	Micro Avg.
		97.4±0.0	81.4±0.0	87.3±0.0	Macro Avg.
CRF	Joint AB-LSTM	86.2±2.6	53.1±1.4	65.7±1.4	⊕
		93.6±0.2	98.8±0.3	96.1±0.2	⊖
		92.6±0.4	93.0±0.3	92.3±0.4	W. Avg.
		93.0±0.3	93.0±0.3	93.0±0.3	Micro Avg.
		89.9±1.3	75.9±0.7	80.9±0.8	Macro Avg.

(b) Hold-out: model inferred with the train and dev sets and evaluated with the test set.

Table 7.5: Results of the best performing approach (dense + Joint AB-LSTM) with the IxaMed-E corpus, evaluated using the gold mentions and the automatic entities obtained with MER.

In addition, we developed the third scenario randomly missing 20%, 40%, 60% and 80% of the entities, yielding the results shown in Figure 7.4. We can observe that as the percentage of missed entities increases, the f-measure of the positive class decreases. Specifically, it follows a quartic polynomial function, which is represented in (7.1) for the dev set and in (7.2) for the test set.

$$f(x) = -3.66 \cdot 10^{-5}x^4 + 6.33 \cdot 10^{-3}x^3 - 3.35 \cdot 10^{-1}x^2 + 4.08x + 79.9 \quad (7.1)$$

$$f(x) = -2.66 \cdot 10^{-5}x^4 + 4.69 \cdot 10^{-3}x^3 - 2.53 \cdot 10^{-1}x^2 + 3.08x + 75.2 \quad (7.2)$$

For instance, when 20% of the entities were dropped randomly, the f-measure of the positive class for the dev set decreased from 79.9 to 72.2. Note that from 20% to the right of Figure 7.4 depicts too pessimistic situations since, current approaches in MER are above 85.75% of f-measure for Spanish and 90.94% of f-measure for English (Lample et al., 2016).

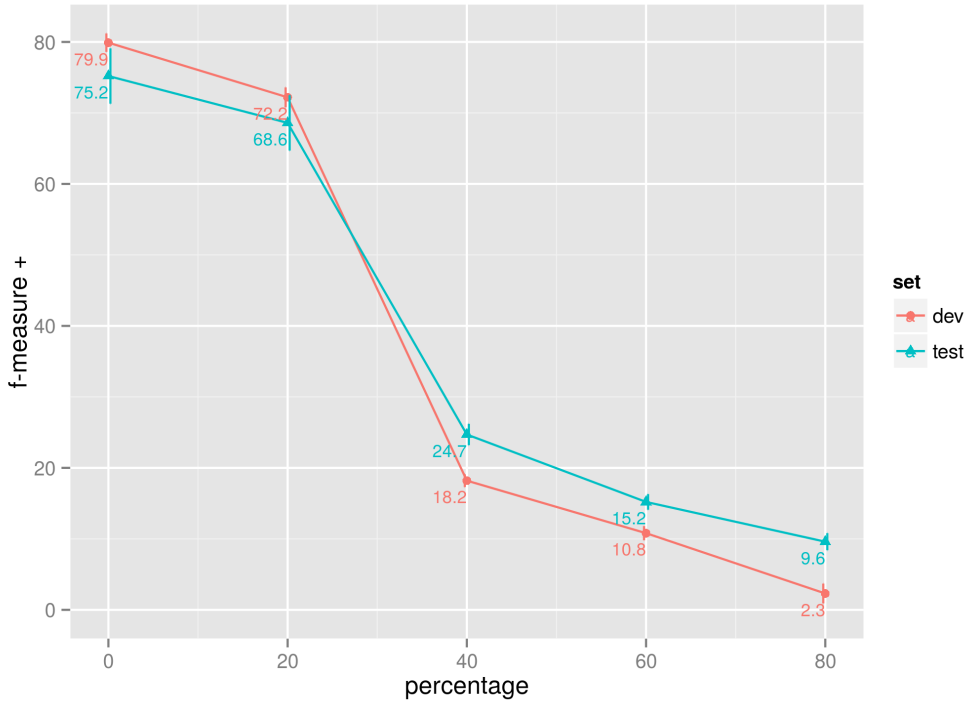


Figure 7.4: Performance of the ADR extraction system with the IxaMed-E corpus, varying the percentage of entities dropped in the evaluation set.

7.3.2 Discussion

When the ideal MER is replaced with the CRF classifier, whenever an entity that takes place in an ADR was not recognized by the CRF (leading to a False Negative entity), an ADR candidate is missed and thus the ADR detection system cannot detect it (yielding a False Negative relation). This happened with around 58% of the undetected ADRs in the dev set and around 80% in the test set. However, the decay in performance from the first scenario to the third one was not as high as one might expect, bearing in mind that the f-measure for the CRF was relatively low (57.4 in the dev set and 57.2 in the test set). Specifically, the performance worsened 5.4 points in the dev set and 10 in the test set. The reason is simple, a big number of the entities missed corresponded to non-ADR relations. The cause of this can be explained, again, with the class imbalance. ADR relations are scarce and missing some entities does not have an impact in the results while they are important in clinical practice. Specifically, the CRF errors caused around 41% of the undetected ADRs in the dev set and 34% in the test set.

In addition, when an entity is incorrectly recognized by the CRF (leading to a False Positive entity), a new ADR candidate is created and the ADR detection can retrieve it as ADR (yielding a False Positive relation). This does not affect to the result given in Table 7.5 because the evaluations are based on the gold annotation made by the experts. Nevertheless, this would produce around 20% of the incorrect ADRs in the dev set and 36% in the test set.

7.3.3 Error analysis

Analyzing the predictions obtained with the MER entities, we can find examples of ADRs that were not found by the system because their entities were not detected in the entity recognition step. For instance, in the sentence given in Figure 7.5 the ADR '*enteropatía - AINEs*' was not detected as ADR by the system because the entity "*enteropatía*" was not recognized by the MER system.

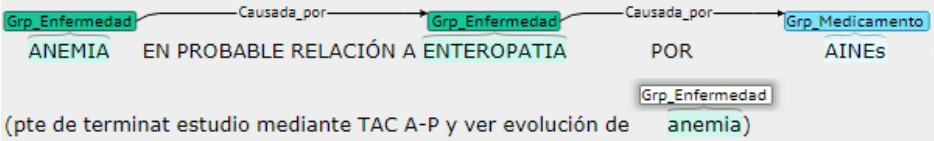


Figure 7.5: Example of sentence in which an ADR was not detected by the system due to the MER errors. The sentence means 'Anemia in probable relation to enteropathy because of NSAIDs (pending to finish study by A-P CT Scan and see evolution of anemia)'.

In these predictions we can also find cases where the prediction was not correct because the drug-disease pair contained an entity that was not labeled by the experts. Then, this ADR did not appear among those annotated by the experts. For example, in the sentence given in Figure 7.6 the word "*anticoagulación*" was detected as drug. Then, it was created the extra pair '*hemoptisis* - *anticoagulación*', which was incorrectly predicted as ADR.

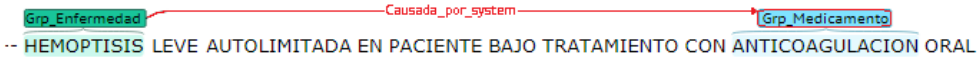


Figure 7.6: Example of sentence in which a drug-disease pair was incorrectly predicted as ADR by the system due to MER errors. The sentence means 'Self-limited slight hemoptysis in patient undergoing treatment with oral anticoagulation'.

7.4 Conclusions

In this chapter, we corroborated that the model inferred with the Joint AB-LSTM yielded the best performance for ADR detection among the three proposed approaches. Next, we found that with the IxaMed-E corpus the performance of the model improved, despite introducing slight variation in the sub-domains, increasing the lexical variability and also the class imbalance. Our impression is that the size of the corpus is relevant to overcome these challenges. Finally, we also evaluated the model with the entities obtained automatically and we observed that the MER errors do not have as high impact on the results as we expected.

Figure 7.7 shows a summary of the aforementioned results, in terms of f-measure of the positive class, for the dev set (Figure 7.7a) and the test set (Figure 7.7b). We can see graphically that the performance improved progressively with each approach and with each corpus and worsened when the ADR candidates were created with the automatic entities obtained with a MER system.

With this experimentation, we answered to the following research questions:

Research Question 7

How do the variations in the size and sub-domains of the corpus affect to the performance of the ADR detection model?

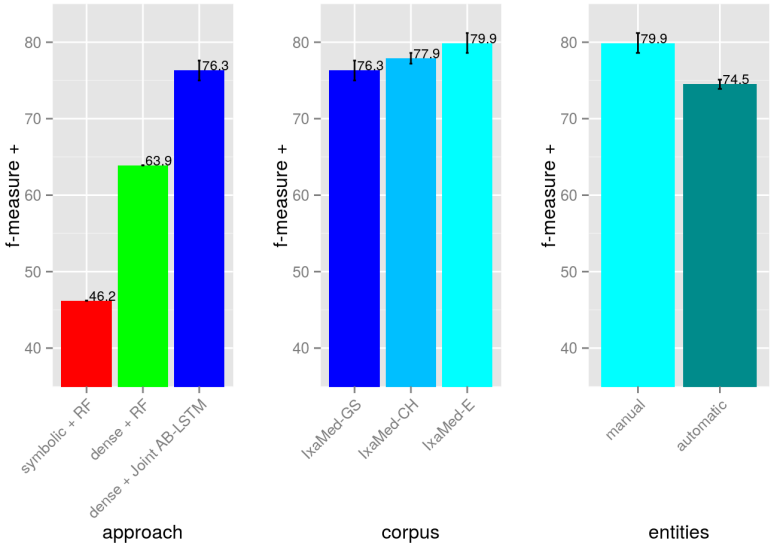
The number of instances used to train the model has a high influence on the detection ability of the ADR extraction. To be precise, the larger the corpus the better the results. In addition, the model can be robust to the different sub-domains introduced by the use of EHRs of different hospitals with different departments. Then, despite the fact that EHRs are difficult to obtain, it is advisable to employ as many as possible.

Research Question 8

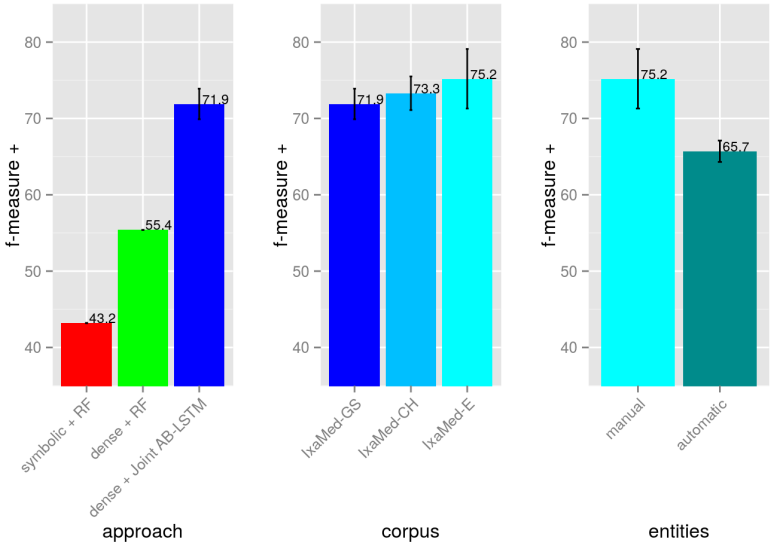
How is the tolerance of the ADR detection model to the noise introduced by the automatic medical entity recognition?

The use of the automatic entities to create the drug-disease pairs worsens the performance of the ADR detection. Mainly, the model inferred in the ADR detection step cannot find the ADRs with entities that were not recognized in the entity recognition step. However, the drop in performance is lower than we expected.

Open questions. Having seen that the neural networks offer a better performance and are more robust against class imbalance, we found of much interest to explore again the extraction of inter-sentence and intra-sentence ADRs with this approach.



(a) F-measure of the positive class obtained for the dev set.



(b) F-measure of the positive class obtained for the test set.

Figure 7.7: Summary of the results obtained in the comparison of the three approaches (symbolic + RF, dense + RF and dense + Joint AB-LSTM'), the three corpus (IxaMed-GS, IxaMed-CH and IxaMed-E) and the two MER systems ("manual" stands for gold mentions and "automatic" for CRF entities). Note that the f-measure is represented from 35.

8 Conclusions and future work

8.1 Summary of the research

In this work we developed ADR extraction, that is, extraction of drug-disease pairs in which the drug caused the disease. We approached it as a relation extraction task, even though some related works defined ADR extraction as the recognition of the caused disease while omitted the causing drug (details about the definition adopted in this work are given in Section 1.2 and the definitions and approaches in related works are given in Section 2.2). We considered the extraction of both inter- and intra-sentence relations, that is to say, relations where the entities are placed in the same sentence or in different sentences. However, considering all the possible relations produced a high class imbalance and the results achieved were poor. Thus, following the main stream, we restricted to intra-sentence relation extraction. Despite of this, there was still an unequal distribution of the class produced because the ADRs are rare events and then, there were four times more negative than positive instances.

We extracted ADRs from EHRs written in Spanish. The EHRs are written using informal language, presenting misspellings as well as standard and non-standard abbreviations (turn to Section 3.2 for more information about the EHRs). This increases the lexical variability present in this type of documents. Furthermore, our work is focused on Spanish, a language with fewer available resources than others such as English. This makes processing these documents difficult (Névéol et al., 2018).

With this framework, we had to tackle some challenges to improve the

ADR detection. Mainly, we had to overcome the class imbalance of the ADR instances and the lexical variability of the EHRs.

Our main objective was to create a model able to detect automatically ADRs in EHRs written in Spanish (see Section 1.3). To this end, we turned to a pipeline approach to find ADR relations and we focused particularly on the ADR detection step than on the MER step (turn to Figure 1.2 to see the scheme of the pipeline). Note that the research works on MER achieved an f-measure of 90%, while research works on relation extraction achieved an f-measure about 70% (Dalianis, 2018). First, we started using symbolic representations of drug-disease pairs with a traditional classifier such as RF. Next, we replaced the symbolic representations by dense representations based on embeddings and improved them with some smoothing techniques. Finally, we resorted to neural networks, a different classification algorithm that is able to infer abstract dense features. In addition, we analyzed the effect of large corpora and automatic entities in the performance of the model.

From the experiments, we concluded that the neural networks, specifically the Joint AB-LSTM, was the best option for ADR detection. Furthermore, the performance was improved using a larger corpora, achieving an f-measure for the positive class of 79.9 in the dev set, 75.2 in the test set and 80.8 with 10-fold cross-validation.

In the rest of this chapter, we give the responses to the main research question together with the concluding remarks (Section 8.2), we present our contributions (Section 8.3), we explain the future lines of the work (Section 8.4), we enumerate the publications derived from this work (Section 8.5) and the intellectual property registry (Section 8.6).

8.2 Concluding remarks

This work was boosted by the following research question:

Main Research Question

How can NLP techniques be applied to aid in the extraction of ADRs in EHRs written in Spanish?

This main research question was broken down into eight research questions that arose and were responded throughout Chapters 4, 5, 6 and 7. In this section we bring together the concluding remarks arisen after thinking over the responses suggested to these research questions.

ADRs are rare events, then, supervised classification algorithms tend to be biased and learning to predict the minority class is complex. The application of approaches to overcome the **class imbalance** improves the performance of the ADR detection model to find **inter-** as well as **intra-sentence** ADRs. However, the results are considerably better in the intra-sentence scope than in the inter-sentence scope.

A key issue in the extraction of ADRs is the operative **characterization** of events. With regard to initial symbolic characterizations, if both inter-sentence and intra-sentence relations are taken into account, features related to the distances between the entities involved result relevant for the task. If the ADR detection is focused on intra-sentence ADRs, the word-forms and the lemmas of the entities and their contexts are more relevant. NLP rapidly evolved towards dense characterizations. Dense representations have the strength of exploiting semantic relatedness in dense low dimensional spaces. This is an important factor in our task to cope with lexical variability. We corroborated that dense representations outperform symbolic ones and it seemed as if the model gains generalization ability.

Another important factor is the **classification approach**. In this work we compared a traditional supervised classification approach (RF) and an emerging technique based on deep neural networks (Joint AB-LSTM) and found that Joint AB-LSTM outperformed RF. We speculated about the reasons behind. An outstanding difference between traditional and neural approaches rests on the generation of the inherent characterization for the instances. While traditional approaches make use of hand-crafted features (either in their symbolic or embedded as dense representations), neural approaches infer, automatically, abstract features. Nevertheless, we found that FFNN did not outperform the RF when the instances were characterized with smoothed embeddings. Our hypothesis to explain that Joint AB-LSTM outperform RF is that the information captured from the context is crucial in relation extraction. While RF exploits the context in a static way, Joint AB-LSTM can leverage the context dynamically. Furthermore, we observed, empirically, that Joint AB-LSTM networks are less sensitive to class imbalance than RF.

Variations in the size and domain of the corpus have an **effect in the performance** of the ADR detection model. To be precise, the larger the corpus the better the results. Regarding the variations associated to different sub-domains introduced by the use of EHRs of different hospitals, Joint AB-LSTM resulted robust. Needless to say, the errors propagated from the MER

step affect the ADR detection. Missing entities lead to undiscovered relations. However, the drop in performance is not as dramatic as we expected and follows a polynomial function of degree 4.

8.3 Contributions

The main contribution of this work is that the ADR extraction was developed using EHRs written in Spanish. To the best of our knowledge, for ADR extraction in texts written in Spanish, we are the first employing EHRs. Other contributions derived from the tasks carried out during this work are:

- **Combination of approaches to tackle the high class imbalance.** (Chapter 4)

  We made a step ahead in the development of NLP methods that deal with ADR extraction defined as relation extraction task between a causative drug and the adverse reaction. As a first approach we tackled both inter- and intra-sentence ADR extraction, even though the main-stream in the related works just focused on intra-sentence relations. In this context, inference algorithms should be suited to cope with the challenge of an extremely high class imbalance (a extremely high number of candidates are unrelated as ADRs). Although the imbalance problem diminishes considerably in intra-sentence scenarios, we explored classical approaches to tackle the class imbalance (sampling, cost-sensitive learning, ensemble learning, one-class classification) in the context of inter- and intra-sentence ADR extraction. We observed that the combination of them, precisely sampling and cost-sensitive learning, was beneficial in our framework.

  Besides, in an attempt to discard non-ADR instances and alleviate the class imbalance we tried, as well, negation extraction. We developed two ways of detecting negated medical entities in EHRs: an adaptation of the NegEx tool and a CRF using dense characterizations. We corroborated, however, that class imbalance can be, somehow, tackled in intra-sentence ADR extraction while there is room for improvement in inter-sentence relation extraction.

- **Mechanisms to deal with lexical variability.** (Chapter 5 and Chapter 6)

  NLP in the medical domain dealing with EHRs has, among others, the challenge of high lexical variability (large specialized vocabularies, non-standard abbreviations, misspellings, etc.) and lack of available corpora. Quantitatively, there is a reflect of the lexical variability in the remarkable ratio of OOV elements. To cope with this issue it results crucial to propose not only competitive inference algorithms but also robust characterizations of the instances. Throughout this work we analyzed two state-of-the-art classification techniques (RF, Joint AB-LSTM) and two alternative representations (symbolic, dense). We experimentally corroborated that context-aware embeddings (dense representations created taking into account the embeddings of the context-words) are useful to preserve the lexical nuances in this domain. In addition, to alleviate the influence that the lack of training samples might have in the quality of the inferred dense representations, we proposed the use of smoothing techniques. Note that while we are using dense representations, applying smoothing techniques yields coarse grained representations. That is, smoothing helps to avoid superficial variations and, hence, makes different (but close) points in the space to be equivalent.

  Moreover, we have observed that dense spaces of lemmas also helped to tackle the lexical variability. In fact, lemmatization was particularly effective in the neural networks used for ADR extraction.

- **Tolerance to external noise.** (Chapter 7)

  We exposed the ADR extraction system to two types of sources of noise. On the one hand, we assessed the impact of corpora from slightly different sources (different hospitals with different services or specializations). On the other hand, we analyzed the influence of miss-recognized medical entities into the ADR detection step leading to a fully automatic ADR extraction system. We corroborated that the Joint AB-LSTM is able to cope with these types of noise although, naturally, there is a small decrease in its performance due to the missed entities involved in the ADR pairs.

8.4 Future work

As a result of this work, we have a model able to extract ADRs following the objectives stated in Section 1.3. However, there are still several points that we would like to tackle as future work:

- **Extract intra-sentence as well as inter-sentence ADRs.**
  Because of the high class imbalance, we focused on ADRs whose entities are in the same sentence. The model inferred with neural networks seemed less sensitive to class imbalance. Then, we would like to find ADRs with entities placed in different sentences using this approach. We could explore two options, incorporating the inter-sentence ADRs as new examples to infer the model or inferring a separate model to detect the inter-sentences ADRs.

- **Develop entity recognition and relation extraction simultaneously.**
  Given that with the pipeline models the errors of the entity recognition are propagated to the relation extraction, we would like to create a neural joint model for entity recognition and relation extraction (Zheng et al., 2016; Li et al., 2017). That is, using neural networks to develop simultaneously the MER and the ADR detection steps.

8.5 Publications

Derived from this work, we have some publications describing the task and the results. There are 5 of them in journals indexed by the Journal Citation Report (JCR): 3 Q1, 1 Q2 and 1 Q3. All the articles are enumerated below in reverse chronological order of their publication together with the Impact Factor (IF):

1. Sara Santiso, Alicia Pérez, and Arantza Casillas. Smoothing dense spaces for improved relation extraction between drugs and adverse reactions *International Journal of Medical Informatics*. Elsevier. ISSN 1386-5056.
   JCR: Q1, IF: 2.957
   [accepted, awaiting publication]

2. Sara Santiso, Arantza Casillas, Alicia Pérez, and Maite Oronoz. Word-embeddings for negation detection in health records written in Spanish. *Soft Computing*, 1–7, 2018. Springer. ISSN 1432-7643. DOI https://doi.org/10.1007/s00500-018-3650-7. URL https://link.springer.com/content/pdf/10.1007%2Fs00500-018-3650-7.pdf.
JCR: Q2, IF: 2.367

3. Sara Santiso, Alicia Pérez, and Arantza Casillas. Exploring Joint AB-LSTM with embedded lemmas for Adverse Drug Reaction discovery. *IEEE Journal of Biomedical and Health Informatics*, 1–8, 2018. IEEE. ISSN 2168-2194. DOI https://doi.org/10.1109/JBHI.2018.2879744. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8523679.
JCR: Q1, IF: 3.850

4. Sara Santiso, Arantza Casillas, and Alicia Pérez. The class imbalance problem detecting Adverse Drug Reactions in Electronic Health Records. *Health Informatics Journal*, 1–11, 2018. SAGE. ISSN 1460-4582. DOI https://doi.org/10.1177/1460458218799470. URL https://journals.sagepub.com/doi/pdf/10.1177/1460458218799470.
JCR: Q3, IF: 1.833

5. Sara Santiso, Arantza Casillas, Alicia Pérez, and Maite Oronoz. Medical entity recognition and negation extraction: Assessment of NegEx on Health Records in Spanish. In *2017 International Work-Conference on Bioinformatics and Biomedical Engineering (IWBBIO)*, pages 177–188, Granada, Spain, April 26-28 2017. Springer. ISSN 0302-9743. DOI https://doi.org/10.1007/978-3-319-56148-6_15. URL https://link.springer.com/chapter/10.1007/978-3-319-56148-6_15.
Lecture Notes in Computer Science (LNCS), Lecture Notes in Bioinformatics (LNBI)

6. Arantza Casillas, Arantza Díaz de Ilarraza, Kike Fernandez, Koldo Gojenola, Maite Oronoz, Alicia Pérez, and Sara Santiso. IXAmed-IE: online medical entity identification and ADR event extraction in Spanish. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 845–849, Shenzhen, China, December 15-18 2016. IEEE. DOI http://doi.ieeecomputersociety.org/10.

1109/BIBM.2016.7822636. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7822636.

7. Arantza Casillas, Alicia Pérez, Maite Oronoz, Koldo Gojenola, and Sara Santiso. Learning to extract adverse drug reaction events from electronic health records in Spanish. *Expert Systems with Applications*, 61:235–245, 2016. Elsevier. ISSN 0957-4174. DOI http://dx.doi.org/10.1016/j.eswa.2016.05.034. URL https://www.sciencedirect.com/science/article/pii/S0957417416302615.
JCR: Q1, IF: 3.768

8. Sara Santiso, Arantza Casillas, Alicia Pérez, Maite Oronoz, and Koldo Gojenola. Document-level adverse drug reaction event extraction on electronic health records in Spanish. *Procesamiento del Lenguaje Natural*, 56:49–56, 2016. Sociedad Española para el Procesamiento del Lenguaje Natural. ISSN 1135-5948. URL http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/5286.
Certificate of Excellence FECYT

9. Sara Santiso, Arantza Casillas, Alicia Pérez, Maite Oronoz, and Koldo Gojenola. Adverse drug event prediction combining shallow analysis and machine learning. In *Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis (Louhi)*, pages 85–89, Gothenburg, Sweden, April 26-30 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W14-1113.

8.6 Intellectual Property Registry

Apart from the publications enumerated in the previous section, we hold the intellectual property registry of a system for automatic entity recognition:

1. Aitziber Atucha, Arantza Casillas, Koldo Gojenola, Maite Oronoz, Alicia Pérez, Olatz Perez de Viñaspre, and Sara Santiso. Sistema automático para la detección automática de entidades del dominio médico en español. Number: SS-23-19. Place: Donostia. Date: 01/18/2019.

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org/. Software available from tensorflow.org.

Agarwal, S. and Yu, H. (2010). Biomedical negation scope detection with conditional random fields. *Journal of the American Medical Informatics Association*, 17(6):696–701.

Akhtyamova, L., Alexandrov, M., and Cardiff, J. (2017). Adverse drug extraction in Twitter data using convolutional neural network. In *28th International Workshop on Database and Expert Systems Applications*, pages 88–92.

Aramaki, E., Miura, Y., Tonoike, M., Ohkuma, T., Masuichi, H., Waki, K., and Ohe, K. (2010). Extraction of adverse drug effects from clinical records. In *MedInfo*, pages 739–743.

Artetxe, M., Labaka, G., and Agirre, E. (2016). Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294.

Attardi, G., Cozza, V., and Sartiano, D. (2015). Annotation and extraction of relations from Italian medical records. In *6th Italian Information Retrieval Workshop*, pages 1–12.

Bach, N. and Badaskar, S. (2007). A review of relation extraction. *Literature review for Language and Statistics II*, 2:1–15.

Ben Abacha, A. and Zweigenbaum, P. (2011). Medical entity recognition: A comparaison of semantic and statistical methods. In *Proceedings of BioNLP 2011 Workshop*, pages 56–64.

Benikova, D., Biemann, C., Kisselew, M., and Pado, S. (2003). GermEval 2014 named entity recognition shared task: companion paper. *Proceedings of the KONVENS GermEval Shared Task on Named Entity Recognition*, 7:1–9.

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

Botsis, T., Nguyen, M. D., Woo, E. J., Markatou, M., and Ball, R. (2011). Text mining for the vaccine adverse event reporting system: medical text classification using informative feature selection. *Journal of the American Medical Informatics Association*, 18(5):631–638.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Campillos, L., Deléger, L., Grouin, C., Hamon, T., Ligozat, A.-L., and Névéol, A. (2018). A French clinical corpus with comprehensive semantic annotations: development of the medical entity and relation limsi an-

notated text corpus (MERLOT). *Language Resources and Evaluation*, 52(2):571–601.

Cardellino, C. (2016). Spanish Billion Words Corpus and Embeddings. http://crscardellino.me/SBWCE/.

Casillas, A., de Ilarraza, A. D., Fernandez, K., Gojenola, K., Oronoz, M., Pérez, A., and Santiso, S. (2016a). IXAmed-IE: On-line medical entity identification and ADR event extraction in Spanish. In *2016 IEEE International Conference on Bioinformatics and Biomedicine*, pages 846–849.

Casillas, A., Pérez, A., Oronoz, M., Gojenola, K., and Santiso, S. (2016b). Learning to extract adverse drug reaction events from electronic health records in Spanish. *Expert Systems with Applications*, 61:235–245.

Celli, F. (2010). UNITN: Part-Of-Speech counting in relation extraction. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 198–201.

Ceusters, W., Elkin, P., and Smith, B. (2007). Negative findings in electronic health records and biomedical ontologies: a realist approach. *International Journal of Medical Informatics*, 76:326–333.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27.

Chapman, W. W., Bridewell, W., Hanbury, P., Cooper, G. F., and Buchanan, B. G. (2001). A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of Biomedical Informatics*, 34(5):301–310.

Chapman, W. W., Hilert, D., Velupillai, S., Kvist, M., Skeppstedt, M., Chapman, B. E., Conway, M., Tharp, M., Mowery, D. L., and Deleger, L. (2013). Extending the NegEx lexicon for multiple languages. *Studies in Health Technology and Informatics*, 192:677–681.

Chen, X., Xu, L., Liu, Z., Sun, M., and Luan, H. (2015). Joint learning of character and word embeddings. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 1236–1242.

Christopoulou, F., Miwa, M., and Ananiadou, S. (2018). A walk-based model on entity graphs for relation extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 81–88.

Cocos, A., Fiks, A. G., and Masino, A. J. (2017). Deep learning for pharmacovigilance: recurrent neural network architectures for labeling adverse drug reactions in Twitter posts. *Journal of the American Medical Informatics Association*, 24(4):813–821.

Cohen, K. B. and Demner-Fushman, D. (2014). *Biomedical Natural Language Processing*. Natural Language Processing. John Benjamins Publishing Company.

Cohen, P. R. (2015). DARPA's Big Mechanism program. *Physical Biology*, 12(4):1–9.

Coloma, P. M., Schuemie, M. J., Trifirò, G., Gini, R., Herings, R., Hippisley-Cox, J., Mazzaglia, G., Giaquinto, C., Corrao, G., Pedersen, L., van der Lei, J., and Sturkenboom, M. (2011). Combining electronic healthcare databases in Europe to allow for large-scale drug safety monitoring: the EU-ADR Project. *Pharmacoepidemiology and Drug Safety*, 20(1):1–11.

Copara, J., Ochoa, J., Thorne, C., and Glavaš, G. (2016). Spanish NER with word representations and conditional random fields. In *Proceedings of the Sixth Named Entity Workshop*, pages 34–40.

Costumero, R., López, F., Gonzalo-Martín, C., Millan, M., and Menasalvas, E. (2014). An approach to detect negation on medical documents in Spanish. In *International Conference on Brain Informatics and Health*, pages 366–375.

Cotik, V., Roller, R., Xu, F., Uszkoreit, H., Budde, K., and Schmidt, D. (2016). Negation detection in clinical reports written in German. In *Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining*, pages 115–124.

Cruz, N. P., Maña, M. J., and Mata, J. (2010). Aprendizaje automático versus expresiones regulares en la detección de la negación y la especulación en biomedicina. *Procesamiento del Lenguaje Natural*, 45(0):77–85.

Cruz, N. P., Maña, M. J., Mata, J., and Pachón, V. (2012). A machine-learning approach to negation and speculation detection in clinical texts. *Journal of the Association for Information Science and Technology*, 63(7):1398–1410.

Cruz, N. P., Morante, R., López, M. J. M., Vázquez, J. M., and Calderón, C. L. P. (2017). Annotating negation in Spanish clinical texts. In *Proceedings of the Workshop Computational Semantics Beyond Events and Roles*, pages 53–58.

Dalianis, H. (2018). *Clinical Text Mining: Secondary Use of Electronic Patient Records*. Springer.

Dalianis, H., Hassel, M., Henriksson, A., and Skeppstedt, M. (2012). Stockholm EPR Corpus: A clinical database used to improve health care. In *Swedish Language Technology Conference*, pages 17–18.

de la Peña, S., Segura-Bedmar, I., Martínez, P., and Martínez, J. L. (2014). ADRSpanishTool: a tool for extracting adverse drug reactions and indications. *Procesamiento del Lenguaje Natural*, 53:177–180.

Deléger, L. and Grouin, C. (2012). Detecting negation of medical problems in French clinical notes. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 697–702.

Deléger, L., Ligozat, A.-L., Grouin, C., Zweigenbaum, P., and Névéol, A. (2014). Annotation of specialized corpora using a comprehensive entity and relation scheme. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 1267–1274.

Demner-Fushman, D., Cohen, K. B., Ananiadou, S., and Tsujii, J., editors (2018). *Proceedings of the SIGBioMed Workshop on Biomedical Natural Language Processing*.

Dessì, D., Fenu, G., Marras, M., and Recupero, D. R. (2018). Bridging learning analytics and cognitive computing for big data classification in

micro-learning video collections. *Computers in Human Behavior*, pages 1–10.

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

Domingos, P. (1999). MetaCost: A general method for making classifiers cost-sensitive. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 155–164.

Duque, A., Martinez-Romo, J., and Araujo, L. (2015). Extracción no supervisada de relaciones entre medicamentos y efectos adversos. *Procesamiento del Lenguaje Natural*, 55:83–90.

Ebrahimi, J. and Dou, D. (2015). Chain based RNN for relation classification. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1244–1249.

Elkan, C. (2001). The foundations of cost-sensitive learning. In *International Joint Conference on Artificial Intelligence*, volume 17, pages 973–978.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2):179–211.

Estabrooks, A., Jo, T., and Japkowicz, N. (2004). A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, 20(1):18–36.

Fabregat, H., Araujo, L., and Martinez-Romo, J. (2018). Deep neural models for extracting entities and relationships in the new RDD corpus relating disabilities and rare diseases. *Computer Methods and Programs in Biomedicine*, 164:121–129.

Farhan, W., Wang, Z., Huang, Y., Wang, S., Wang, F., and Jiang, X. (2016). A predictive model for medical events based on contextual embedding of temporal sequences. *JMIR Medical Informatics*, 4(4):1–13.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874.

Feng, X., Huang, L., Tang, D., Ji, H., Qin, B., and Liu, T. (2016). A language-independent neural network for event detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 66–71.

Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Thirteenth International Conference on Machine Learning*, volume 96, pages 148–156.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86–92.

Friedrich, S. and Dalianis, H. (2015). Adverse drug event classification of health records using dictionary based pre-processing and machine learning. In *Proceedings of the Sixth International Workshop on Health Text Mining and Information Analysis*, pages 121–130.

Gábor, K., Buscaldi, D., Schumann, A.-K., QasemiZadeh, B., Zargayouna, H., and Charnois, T. (2018). SemEval-2018 Task 7: Semantic relation extraction and classification in scientific papers. In *Proceedings of International Workshop on Semantic Evaluation*.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., and Herrera, F. (2012). A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):463–484.

Ginn, R., Pimpalkhute, P., Nikfarjam, A., Patki, A., O'Connor, K., Sarker, A., Smith, K., and Gonzalez, G. (2014). Mining Twitter for adverse drug reaction mentions: a corpus and classification benchmark. In *Proceedings of the Fourth Workshop on Building and Evaluating Resources for Health and Biomedical Text Processing*, pages 1–8.

Goldberg, Y. and Hirst, G. (2017). *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers.

Gormley, M. R., Yu, M., and Dredze, M. (2015). Improved relation extraction with feature-rich compositional embedding models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1774–1784.

Grishman, R. (2003). Information extraction. *The Handbook of Computational Linguistics and Natural Language Processing*, pages 515–530.

Gupta, S., Pawar, S., Ramrakhiyani, N., Palshikar, G. K., and Varma, V. (2018). Semi-supervised recurrent neural network for adverse drug reaction mention extraction. *BMC Bioinformatics*, 19(8):1–7.

Gurulingappa, H., Fluck, J., Hofmann-Apitius, M., and Toldo, L. (2011). Identification of adverse drug event assertive sentences in medical case reports. In *First International Workshop on Knowledge Discovery and Health Care Management, European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 16–27.

Gurulingappa, H., Mateen-Rajpu, A., and Toldo, L. (2012a). Extraction of potential adverse drug events from medical case reports. *Journal of Biomedical Semantics*, 3(1):15.

Gurulingappa, H., Rajput, A. M., Roberts, A., Fluck, J., Hofmann-Apitius, M., and Toldo, L. (2012b). Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of Biomedical Informatics*, 45(5):885–892.

Habibi, M., Weber, L., Neves, M., Wiegandt, D. L., and Leser, U. (2017). Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14):37–48.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. (2009). The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1):10–18.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

Hazell, L. and Shakir, S. A. (2006). Under-reporting of adverse drug reactions. *Drug Safety*, 29(5):385–396.

He, B., Guan, Y., and Dai, R. (2019). Classifying medical relations in clinical text via convolutional neural networks. *Artificial intelligence in medicine*, 93:43–49.

He, H. and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284.

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L., and Szpakowicz, S. (2010). SemEval-2010 Task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38.

Henriksson, A., Kvist, M., Dalianis, H., and Duneld, M. (2015a). Identifying adverse drug event information in clinical notes with distributional semantic representations of context. *Journal of Biomedical Informatics*, 57:333–349.

Henriksson, A., Zhao, J., Boström, H., and Dalianis, H. (2015b). Modeling electronic health records in ensembles of semantic spaces for adverse drug event detection. In *Proceedings of the 2015 IEEE International Conference on Bioinformatics and Biomedicine*, pages 343–350.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Huynh, T., He, Y., Willis, A., and Rüger, S. (2016). Adverse drug reaction classification with deep neural networks. In *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, pages 877–887.

Intxaurrondo, A., Marimon, M., González-Agirre, A., López-Martín, J. A., Rodriguez, H., Santamaría, J., Villegas, M., and Krallinger, M. (2018). Finding mentions of abbreviations and their definitions in Spanish clinical cases: The BARR2 shared task evaluation results. In *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages co-located with 34th Conference of the Spanish Society for Natural Language Processing*, pages 280–289.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456.

ISO, T. (2005). 20514: 2005 health informatics-electronic health record definition, scope and context standard. *International Organization for Standardization (ISO), Geneva Switzerland*.

Jacobson, O. and Dalianis, H. (2016). Applying deep learning on electronic health records in Swedish to predict healthcare-associated infections. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 191–195.

Jagannatha, A. N. and Yu, H. (2016a). Bidirectional RNN for medical event detection in electronic health records. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 473–482.

Jagannatha, A. N. and Yu, H. (2016b). Structured prediction models for RNN based sequence labeling in clinical text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 856–865.

Jiménez-Zafra, S. M., Díaz, N. P. C., Morante, R., and Martín-Valdivia, M. T. (2019). Neges 2018: Workshop on negation in spanish. *Procesamiento del Lenguaje Natural*, 62:21–28.

Ju, M., Miwa, M., and Ananiadou, S. (2018). A neural layered model for nested named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1446–1459.

Jurafsky, D. and Martin, J. H. (2018). *Speech and language processing*, volume 3. Pearson London.

Kang, T., Zhang, S., Xu, N., Wen, D., Zhang, X., and Lei, J. (2017). Detecting negation and scope in Chinese clinical notes using character and word embedding. *Computer Methods and Programs in Biomedicine*, 140:53–59.

Karimi, S., Metke-Jimenez, A., Kemp, M., and Wang, C. (2015). CADEC: A corpus of adverse drug event annotations. *Journal of Biomedical Informatics*, 55:73–81.

Karlsson, I., Zhao, J., Asker, L., and Boström, H. (2013). Predicting adverse drug events by analyzing electronic patient records. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 125–129.

Katiyar, A. and Cardie, C. (2017). Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 917–928.

Kazama, J., Makino, T., Ohta, Y., and Tsujii, J. (2002). Tuning support vector machines for biomedical named entity recognition. In *Proceedings of the ACL-02 Workshop on Natural Language Processing in the Biomedical Domain*, pages 1–8.

Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, volume 5, pages 1–15.

Kudo, T. (2005). CRF++: Yet another CRF toolkit. http://crfpp.sourceforge.net. Software available at http://crfpp. sourceforge. net.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, volume 1, pages 282–289.

Lai, S., Liu, K., He, S., and Zhao, J. (2016). How to generate a good word embedding. *IEEE Intelligent Systems*, 31(6):5–14.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270.

Lamy, M., Pereira, R., Ferreira, J. C., Melo, F., and Velez, I. (2018). Extracting clinical knowledge from electronic medical records. *Extracting clinical knowledge from electronic medical records*, (3):488–493.

Lavelli, A., Minard, A.-L., and Rinaldi, F., editors (2018). *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis (Louhi)*.

Le, H.-Q., Can, D.-C., Vu, S. T., Dang, T. H., Pilehvar, M. T., and Collier, N. (2018). Large-scale exploration of neural relation classification architectures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2266–2277.

Leaman, R., Khare, R., and Lu, Z. (2015). Challenges in clinical natural language processing for automated disorder normalization. *Journal of Biomedical Informatics*, 57:28–37.

Lecun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Lee, H.-j. and Cho, S. (2006). The novelty detection approach for different degrees of class imbalance. In *International Conference on Neural Information Processing*, pages 21–30.

Lee, K., Qadir, A., Hasan, S. A., Datla, V., Prakash, A., Liu, J., and Farri, O. (2017). Adverse drug event detection in tweets with semi-supervised convolutional neural networks. In *Proceedings of the 26th International Conference on World Wide Web*, pages 705–714.

Legrand, J., Toussaint, Y., Raïssi, C., and Coulet, A. (2018). Syntax-based transfer learning for the task of biomedical relation extraction. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 149–159.

Li, F., Ji, D., Wei, X., and Qian, T. (2015). A transition-based model for jointly extracting drugs, diseases and adverse drug events. In *2015 IEEE International Conference on Bioinformatics and Biomedicine*, pages 599–602.

Li, F., Zhang, M., Fu, G., and Ji, D. (2017). A neural joint model for entity and relation extraction from biomedical text. *BMC Bioinformatics*, 18(1):1–11.

Li, J., Zhou, G., Wang, H., and Zhu, Q. (2010). Learning the scope of negation via shallow semantic parsing. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 671–679.

Li, Q., Deleger, L., Lingren, T., Zhai, H., Kaiser, M., Stoutenborough, L., Jegga, A. G., Cohen, K. B., and Solti, I. (2013). Mining FDA drug labels for medical conditions. *BMC Medical Informatics and Decision Making*, 13(1):1–11.

Limsopatham, N. and Collier, N. H., editors (2016). *Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining*.

Lin, W.-S., Dai, H.-J., Jonnagaddala, J., Chang, N.-W., Jue, T. R., Iqbal, U., Shao, J. Y.-H., Chiang, I.-J., and Li, Y.-C. (2015). Utilizing different word representation methods for Twitter data in adverse drug reactions extraction. In *2015 Conference on Technologies and Applications of Artificial Intelligence*, pages 260–265.

Lin, Y.-F., Tsai, T.-H., Chou, W.-C., Wu, K.-P., Sung, T.-Y., and Hsu, W.-L. (2004). A maximum entropy approach to biomedical named entity recognition. In *Proceedings of the 4th International Conference on Data Mining in Bioinformatics*, pages 56–61.

Lindquist, M. (2008). VigiBase, the WHO global ICSR database system: basic facts. *Drug Information Journal*, 42(5):409–419.

Ling, W., Dyer, C., Black, A. W., and Trancoso, I. (2015). Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1299–1304.

Llanos, L. C. and Ueda, H. (2015). Frecuencia y dispersión léxicas en textos médicos divulgativos en español. *Ibérica*, 30:61–84.

Luo, Y. (2017). Recurrent neural networks for classifying relations in clinical notes. *Journal of Biomedical Informatics*, 72:85–95.

Luo, Y., Thompson, W. K., Herr, T. M., Zeng, Z., Berendsen, M. A., Jonnalagadda, S. R., Carson, M. B., and Starren, J. (2017). Natural language

processing for EHR-based pharmacovigilance: a structured review. *Drug Safety*, 40(11):1075–1089.

Manning, C. D., Manning, C. D., and Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT press.

Marimon, M., Vivaldi, J., and Bel, N. (2017). Annotation of negation in the IULA Spanish clinical record corpus. In *Proceedings of the Workshop Computational Semantics Beyond Events and Roles*, pages 43–52.

Martí, M. A., Martín-Valdivia, M. T., Taulé, M., Jiménez-Zafra, S. M., Nofre, M., and Marsó, L. (2016). La negación en español: análisis y tipología de patrones de negación. *Procesamiento del Lenguaje Natural*, 57:41–48.

Masino, A. J., Forsyth, D., and Fiks, A. G. (2018). Detecting adverse drug reactions on Twitter with convolutional neural networks and word embedding features. *Journal of Healthcare Informatics Research*, 2(1-2):25–43.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. In *Proceedings of Workshop at International Conference on Learning Representations*, pages 1–12.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Ministerio de Energía, Turismo y Agenda Digital (2015). Plan de impulso de las tecnologías del lenguage. http://www.agendadigital.gob.es/tecnologias-lenguaje/Bibliotecaimpulsotecnologiaslenguaje/Detalle%20del%20Plan/Plan-Impulso-Tecnologias-Lenguaje.pdf.

Ministerio de Sanidad y Consumo (2006). Estudio nacional sobre los efectos adversos ligados a la hospitalización (ENEAS). http://www.seguridaddelpaciente.es/resources/contenidos/castellano/2006/ENEAS.pdf.

Miura, Y., Aramaki, E., Ohkuma, T., Tonoike, M., Sugihara, D., Masuichi, H., and Ohe, K. (2010). Adverse-effect relations extraction from massive clinical records. In *Proceedings of the Second Workshop on NLP Challenges in the Information Explosion Era*, pages 75–83.

Miwa, M. and Bansal, M. (2016). End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116.

Morante, R. and Blanco, E. (2012). * SEM 2012 Shared Task: Resolving the scope and focus of negation. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 265–274.

Morante, R. and Daelemans, W. (2009). A metalearning approach to processing the scope of negation. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 21–29.

Morante, R. and Daelemans, W. (2012). Annotating modality and negation for a machine reading evaluation. In *2012 Conference and Labs of the Evaluation Forum-Question Answering For Machine Reading Evaluation*, pages 1–11.

Morante, R., Liekens, A., and Daelemans, W. (2008). Learning the scope of negation in biomedical texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 715–724.

Moreno, I., Boldrini, E., Moreda, P., and Romá-Ferri, M. T. (2017). DrugSemantics: A corpus for named entity recognition in Spanish summaries of product characteristics. *Journal of Biomedical Informatics*, 72:8 – 22.

Moreno-Sandoval, A. and Campillos-Llanos, L. (2013). Design and annotation of MultiMedica - a multilingual text corpus of the biomedical domain. *Procedia - Social and Behavioral Sciences*, 95:33 – 39.

Nakov, P. and Zesch, T. (2014). Proceedings of the 8th international workshop on semantic evaluation. pages 1–20.

Nanni, L., Fantozzi, C., and Lazzarini, N. (2015). Coupling different methods for overcoming the class imbalance problem. *Neurocomputing*, 158:48–61.

Névéol, A., Dalianis, H., Velupillai, S., Savova, G., and Zweigenbaum, P. (2018). Clinical natural language processing in languages other than English: opportunities and challenges. *Journal of Biomedical Semantics*, 9(1):1–13.

Nguyen, T. H. and Grishman, R. (2015). Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48.

Nikfarjam, A., Sarker, A., O'Connor, K., Ginn, R., and 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(3):671–681.

Oronoz, M., Casillas, A., Gojenola, K., and Pérez, A. (2013). Automatic annotation of medical records in Spanish with disease, drug and substance names. In *Iberoamerican Congress on Pattern Recognition*, pages 536–543.

Oronoz, M., Gojenola, K., Pérez, A., Díaz de Ilarraza, A., and Casillas, A. (2015). On the creation of a clinical gold standard corpus in Spanish: Mining adverse drug reactions. *Journal of Biomedical Informatics*, 56:318–332.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318.

Patki, A., Sarker, A., Pimpalkhute, P., Nikfarjam, A., Ginn, R., O'Connor, K., Smith, K., and Gonzalez, G. (2014). Mining adverse drug reaction signals from social media: going beyond extraction. *Proceedings of BioLINK SIG*, 2014:1–8.

Patrick, J. and Wang, Y. (2005). Biomedical named entity recognition system. In *Proceedings of the Tenth Australasian Document Computing Symposium*, pages 1–8.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pennington, J., Socher, R., and Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing*, pages 1532–1543.

Pérez, A., Weegar, R., Casillas, A., Gojenola, K., Oronoz, M., and Dalianis, H. (2017). Semi-supervised medical entity recognition: A study on Spanish and Swedish clinical corpora. *Journal of biomedical informatics*, 71:16–30.

Perotte, A., Pivovarov, R., Natarajan, K., Weiskopf, N., Wood, F., and Elhadad, N. (2014). Diagnosis code assignment: models and evaluation metrics. *Journal of the American Medical Informatics Association*, 21(2):231–237.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 2227–2237.

Piad-Morffis, A., Gutiérrez, Y., and Muñoz, R. (2019). A corpus to support ehealth knowledge discovery technologies. *Journal of biomedical informatics*, pages 1–42.

Qi, Y. (2012). Random forest for bioinformatics. *Ensemble Machine Learning: Methods and Applications*, pages 307–323.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.

Raj, D., Sahu, S., and Anand, A. (2017). Learning local and global contexts using a convolutional recurrent network model for relation classification in biomedical text. In *Proceedings of the 21st Conference on Computational Natural Language Learning*, pages 311–321.

Ren, F., Zhou, D., Liu, Z., Li, Y., Zhao, R., Liu, Y., and Liang, X. (2018). Neural relation classification with text descriptions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1167–1177.

Roller, R., Uszkoreit, H., Xu, F., Seiffe, L., Mikhailov, M., Staeck, O., Budde, K., Halleck, F., and Schmidt, D. (2016). A fine-grained corpus annotation schema of german nephrology records. In *Proceedings of the Clinical Natural Language Processing Workshop*, pages 69–77.

Sahu, S. K. and Anand, A. (2018). Drug-drug interaction extraction from biomedical texts using long short-term memory network. *Journal of Biomedical Informatics*, 86:15–24.

Santiso, S., Casillas, A., and Pérez, A. (2018a). The class imbalance problem detecting adverse drug reactions in electronic health records. *Health Informatics Journal*, pages 1–11.

Santiso, S., Casillas, A., Pérez, A., and Oronoz, M. (2017). Medical entity recognition and negation extraction: Assessment of NegEx on health records in Spanish. In *International Work-Conference on Bioinformatics and Biomedical Engineering*, pages 177–188.

Santiso, S., Casillas, A., Pérez, A., and Oronoz, M. (2018b). Word embeddings for negation detection in health records written in Spanish. *Soft Computing*, pages 1–7.

Santiso, S., Casillas, A., Perez, A., Oronoz, M., and Gojenola, K. (2014). Adverse drug event prediction combining shallow analysis and machine learning. In *Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis (Louhi)*, pages 85–89.

Santiso, S., Casillas, A., Pérez, A., Oronoz, M., and Gojenola, K. (2016). Document-level adverse drug reaction event extraction on electronic health records in Spanish. *Procesamiento del Lenguaje Natural*, (56):49–56.

Santiso, S., Pérez, A., and Casillas, A. (2018c). Exploring joint ab-lstm with embedded lemmas for adverse drug reaction discovery. *IEEE Journal of Biomedical and Health Informatics*, pages 1–8.

Sarker, A. and Gonzalez, G. (2015). Portable automatic text classification for adverse drug reaction detection via multi-corpus training. *Journal of Biomedical Informatics*, 53:196–207.

Sarker, A., Nikfarjam, A., and Gonzalez, G. (2016). Social media mining shared task workshop. In *Biocomputing 2016: Proceedings of the Pacific Symposium*, pages 581–592.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Segura-Bedmar, I., de la Peña, S., and Martınez, P. (2014a). Extracting drug indications and adverse drug reactions from Spanish health social media. In *Proceedings of BioNLP*, pages 98–106.

Segura-Bedmar, I., Martínez, P., Revert, R., and Moreno-Schneider, J. (2015). Exploring Spanish health social media for detecting drug effects. *BMC Medical Informatics and Decision Making*, 15(2):1–9.

Segura-Bedmar, I., Revert, R., and Martínez, P. (2014b). Detecting drugs and adverse events from Spanish social media streams. In *Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis (Louhi)*, pages 106–115.

Settles, B. (2004). Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint workshop on Natural Language Processing in Biomedicine and its Applications*, pages 107–110.

Simonoff, J. S. (2012). *Smoothing methods in statistics*. Springer Science & Business Media.

Skeppstedt, M. (2010). Negation detection in Swedish clinical text. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2010 Second Louhi Workshop on Text and Data Mining of Health Documents*, pages 15–21.

Skeppstedt, M. (2011). Negation detection in Swedish clinical text: An adaption of NegEx to Swedish. *Journal of Biomedical Semantics*, 2(3):1–12.

Skeppstedt, M., Dalianis, H., and Nilsson, G. H. (2011). Retrieving disorders and findings: Results using SNOMED CT and NegEx adapted for Swedish. In *Proceedings of Louhi 2011 Third International Workshop on Health Document Text Mining and Information AnalysisBled*, pages 11–17.

Sohn, S., Kocher, J.-P. A., Chute, C. G., and Savova, G. K. (2011). Drug side effect extraction from clinical narratives of psychiatry and psychology patients. *Journal of the American Medical Informatics Association*, 18(Supplement_1):144–149.

Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437.

Stanovsky, G., Gruhl, D., and Mendes, P. (2017). Recognizing mentions of adverse drug reaction in social media using knowledge-infused recurrent models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 142–151.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., and Tsujii, J. (2012). BRAT: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107.

Straková, J., Straka, M., and Hajič, J. (2016). Neural networks for featureless named entity recognition in Czech. In *International Conference on Text, Speech, and Dialogue*, pages 173–181.

Suárez-Paniagua, V. and Segura-Bedmar, I. (2018). Evaluation of pooling operations in convolutional architectures for drug-drug interaction extraction. *BMC Bioinformatics*, 19(8):209.

Tang, Z., Jiang, L., Yang, L., Li, K., and Li, K. (2015). CRFs based parallel biomedical named entity recognition algorithm employing MapReduce framework. *Cluster Computing*, 18(2):493–505.

Tjong Kim Sang, E. F. (2002). Introduction to the CoNLL-2002 Shared Task: Language-independent named entity recognition. In *Proceedings of the 6th Conference on Natural Language Learning*, pages 1–4.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the CoNLL-2003 Shared Task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning*, pages 142–147.

Tutubalina, E. and Nikolenko, S. (2017). Combination of deep recurrent neural networks and conditional random fields for extracting adverse drug reactions from user reviews. *Journal of Healthcare Engineering*, 2017:1–9.

Uzuner, Ö., South, B. R., Shen, S., and DuVall, S. L. (2011). 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

Van Mulligen, E. M., Fourrier-Reglat, A., Gurwitz, D., Molokhia, M., Nieto, A., Trifiro, G., Kors, J. A., and Furlong, L. I. (2012). The EU-ADR corpus: annotated drugs, diseases, targets, and their relationships. *Journal of Biomedical Informatics*, 45(5):879–884.

Velupillai, S. and Kvist, M. (2012). Fine-grained certainty level annotations used for coarser-grained e-health scenarios. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 450–461.

Vincze, V., Szarvas, G., Farkas, R., Móra, G., and Csirik, J. (2008). The bioscope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics*, 9(11):1–9.

Wang, X., Yang, C., and Guan, R. (2018a). A comparative study for biomedical named entity recognition. *International Journal of Machine Learning and Cybernetics*, 9(3):373–382.

Wang, Y., Liu, S., Afzal, N., Rastegar-Mojarad, M., Wang, L., Shen, F., Kingsbury, P., and Liu, H. (2018b). A comparison of word embeddings for the biomedical natural language processing. *Journal of Biomedical Informatics*, 87:12–20.

Weegar, R., Kvist, M., Sundström, K., Brunak, S., and Dalianis, H. (2015). Finding cervical cancer symptoms in Swedish clinical text using a machine learning approach and NegEx. In *AMIA Annual Symposium Proceedings*, volume 2015, pages 1296–1305.

Weegar, R., Pérez, A., Casillas, A., and Oronoz, M. (2018). Deep medical entity recognition for Swedish and Spanish. In *2018 IEEE International Conference on Bioinformatics and Biomedicine*, pages 1595–1601.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, 5(2):241–259.

World Health Organization (2002a). The importance of pharmacovigilance. pages 1–52.

World Health Organization (2002b). Safety of medicines: a guide to detecting and reporting adverse drug reactions: why health professionals need to take action. pages 1–16.

World Health Organization (2003). ATC/DDD Index 2003. *Guidelines for ATC classification and DDD assignment 2003*.

Wunnava, S., Qin, X., Kakar, T., Rundensteiner, E. A., and Kong, X. (2018). Bidirectional LSTM-CRF for adverse drug event tagging in electronic health records. *Proceedings of Machine Learning Research*, 90:48–56.

Zeng, D., Liu, K., Lai, S., Zhou, G., and Zhao, J. (2014). Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344.

Zhang, Z., Nie, J., and Zhang, X. (2016). An ensemble method for binary classification of adverse drug reactions from social media. In *Proceedings of the Social Media Mining Shared Task Workshop at the Pacific Symposium on Biocomputing*, pages 1–5.

Zhao, J., Henriksson, A., Asker, L., and Boström, H. (2014). Detecting adverse drug events with multiple representations of clinical measurements. In *2014 IEEE International Conference on Bioinformatics and Biomedicine*, pages 536–543.

Zhao, J., Henriksson, A., Asker, L., and Boström, H. (2015). Predictive modeling of structured electronic health records for adverse drug event detection. *BMC Medical Informatics and Decision Making*, 15(4):1–15.

Zheng, S., Xu, J., Zhou, P., Bao, H., Qi, Z., and Xu, B. (2016). A neural network framework for relation extraction: Learning entity semantic and relation pattern. *Knowledge-Based Systems*, 114:12–23.

Zhou, D., Zhong, D., and He, Y. (2014). Biomedical relation extraction: from binary to complex. *Computational and Mathematical Methods in Medicine*, 2014:1–18.

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., and Xu, B. (2016). Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212.

Zubke, M. (2017). Classification based extraction of numeric values from clinical narratives. In *Proceedings of the Biomedical NLP Workshop associated with Recent Advances in Natural Language Processing 2017*, pages 24–31.

A

Negated Medical Entity Recognition

A.1 Introduction

The negation is a linguistic phenomenon that inverts the truth value of a sentence or clause. Consequently, the identification of negated entities is crucial for automatic systems to interpret the information correctly. However, sometimes we do not pay enough attention to it.

The negation is present in all languages and is relatively frequent. A prove of this is that in biomedical texts such as the BioScope corpus, about 13% of the sentences contain negated statements (Vincze et al., 2008). Then, it is important to take into account this for the creation of automatic information extraction systems. Above all in medical records, which are used by the doctor to express impressions, hypothesized explanations of experimental results or negative findings (Li et al., 2010). According to Ceusters et al. (2007), substantial fraction of the observations made by clinicians and entered into patient records are expressed by means of negation or by using terms which contain negative qualifiers.

In the last years, the interest on negation detection has increased. For example, the *SEM Shared Task 2012* (Morante and Blanco, 2012) was aimed at resolving the scope and focus of negation and the *SEPLN 2017 conference* and *SEPLN 2018 conference* (Jiménez-Zafra et al., 2019) included a workshop about Spanish Negation (NEGES).

To tackle the negation we distinguished between the negation trigger word and the scope. The negation trigger word is the cue that indicates the negation, these cues can be nouns ('inability'), verbs ('prevents'), prepo-

sitions ('without'), adverbs ('never'), determiners ('no'), pronouns ('none'), prefixes ('unsolved') or conjunctions ('neither ... no') (Morante and Daelemans, 2012). The scope is the part of the text affected by the negation. Inside the scope there is the focus, the word that is negated explicitly (Martí et al., 2016).

In this work we explored two different approaches for negation detection, one based on regular expressions using NegEx (Chapman et al., 2001) and another based on machine learning using CRF (Lafferty et al., 2001). In this case, the scope of the negation is restricted to the medical entities, that is to say, we are only interested on the detection of the negated entities and not on the detection of other type of negated information. The interest in negation detection is that it could be used to discard those pairs that contain negated entities, or as a feature to represent the drug-disease pairs in the ADR extraction task, as it was explained in Chapter 4.

The rest of the appendix is organized as follows: Section A.2 describes previous works on detection of negated entities. Section A.3 explains the approach based on NegEx and shows the results obtained in the experiments. Section A.4 explains the approach based on CRF and shows the results obtained in the experiments. Finally, Section A.5 is devoted to present the conclusions.

A.2 Related work

Given that we explored two approaches for the detection of the negated entities, we differentiate those related works that employed rule-based approaches (specifically, NegEx) and those that employed machine learning methods.

The **NegEx** algorithm was developed by Chapman et al. (2001) to detect the negation of findings and diseases in narrative medical records in English. Observing that NegEx was used for texts written in other languages different from English, Chapman et al. (2013) decided to create a shareable lexicon for NegEx in order to facilitate its use in other languages (Swedish, French and German). To do this, the negation triggers were translated by research groups and they were represented using lexical ontologies.

Skeppstedt (2010) and Skeppstedt et al. (2011) adapted NegEx to Swedish clinical texts (EHRs). Skeppstedt (2010) used the Swedish translation of ICD-10 codes for symptoms and diseases and Skeppstedt et al. (2011) used the terms of the Swedish translation of SNOMED CT belonging to the semantic

categories "finding" or "disorder". In both cases, the negation trigger words were translated to English and expanded. Weegar et al. (2015) employed the adaptation of NegEx to Swedish in order to find cervical cancer symptoms.

Deléger and Grouin (2012) adapted Negex to French to detect the negation in clinical reports in cardiology and Cotik et al. (2016) adapted Negex to German to detect the negation in discharge summaries and clinical notes of the nephrology domain. In both cases, the adaptation was done by translating the negation phrases to the corresponding language.

For Spanish, Costumero et al. (2014) adapted NegEx to detect negated entities in medical documents written in Spanish. To this end, the NegEx algorithm was not changed, but the terms used as negation triggers.

Regarding the **machine learning** approaches for the detection of the negated entities, Cruz et al. (2010) proved that these approaches were able to outperform approaches based on regular expressions. Specifically, the authors focused on the detection of the negation expressions and used the C4.5 and NB algorithms.

Before the aforementioned work, Morante et al. (2008) already divided the negation detection in two phases: i) identification of the negation signals and ii) determination of the negation scope, using k-Nearest Neighbors in both of them. Next, Morante and Daelemans (2009) employed different classifiers for each phase. For the negation signal, the IGTREE algorithm was used and for the negation scope, the CRF algorithm was used a metalearner with three base classifiers: i) Memory-based learning , ii) SVM, and iii) CRF. After that, Agarwal and Yu (2010) and Cruz et al. (2012) also detected the negated entities with two phases. Agarwal and Yu (2010) used CRF in both of them and Cruz et al. (2012) used C4.5 and SVM also in both of them.

The main features used by Morante et al. (2008) and Morante and Daelemans (2009) for the first phase (identification of the negation signals) were the word-form, lemma, POS and chunk of the token and its context. For the second phase (determination of the negation scope) the main features were the word-form, POS and chunk of the negation signal, the paired token, the tokens between the negation signal and the token in focus. The features used in both phases by Cruz et al. (2012) also contain information about the signal, the paired token, their context or the tokens in between such as the lemma or POS. However, Agarwal and Yu (2010),replaced the non-cue words by the POS tag and the cue word by the tag 'CUE' for the negation scope detection.

Apart from this, we found that Kang et al. (2017) also used dense rep-

resentations. Specifically, the authors incorporated word-embeddings and character-embeddings to a CRF algorithm to detect the negation in admission notes and discharge summaries, but these were written in Chinese.

In the aforementioned works that detected negated entities using machine learning approaches, the authors employed the BioScope corpus, with the exception of Kang et al. (2017), that used admission notes and discharge summaries written in Chinese.

A.3 Negation detection with NegEx

The first approach used for the detection of the negated medical entities was the rule-based system NegEx (Chapman et al., 2001). According to the creators of NegEx, "NegEx is a simple algorithm that could be implemented quickly and easily to determine whether an indexed term is negated" (Chapman et al., 2001). However, it was already used to detect the negation in EHRs yielding good results. For this reason, we decided to use this algorithm for the task.

Negex identifies the UMLS terms in the text and later it searches the trigger words that indicate negation, labeling as negated entities those inside a token-window near the negation trigger word. This means that the task is divided in two steps: i) it is developed the detection of the entities and ii) it is determined if these entities are negated or not.

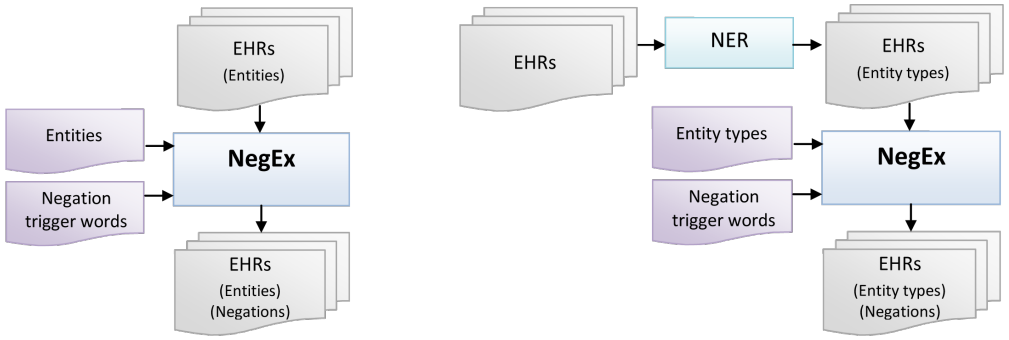
A.3.1 NegEx adaptation

NegEx preprocesses the text by individual sentences using exact-match with respect to two lists, one for gathering the medical terms from UMLS and another one for listing the negation trigger words. This supposes a limitation mainly for the NER stage given that the lists have to be modified when we want to add new entities or we use NegEx in other domains and languages. Otherwise, some entities are not recognized and, therefore, we can not detect their negations. We consider negation detection the more robust part in NegEx as although it also uses exact-match, the negation trigger words are common in different types of texts.

Motivated by this, we made an adaptation that makes possible to apply different techniques for the entity recognition and not restrict to the simply exact-matching integrated in NegEx. This adaptation consists in replacing

each medical entity (e.g. drugs and diseases) by a reserved word that correspond to the entity type. For example, for the sentence '*no haber presentado hipoglucemias hasta el momento actual*' (meaning 'it has not presented hypoglycemias until the current moment'), the input of NegEx would be '*no haber presentado Grp_Enfermedad hasta el momento actual*' ("Grp_Enfermedad" would be the word reserved for the diseases).

Figure A.1 shows the workflow of the original version of NegEx (Figure A.1a) and the workflow of our adaptation (Figure A.1b). In our adaptation the entities can be recognized in a previous step using different NER approaches and they are replaced with the corresponding entity type. After that, the NegEx algorithm is used as in the original version, but with a short list of the words reserved for the entity types instead of a list of entities. Finally, those entities that appear negated in the records are obtained.



(a) Original NegEx. (b) NegEx adaptation.

Figure A.1: Workflow of NegEx before and after our adaptation.

In order to see the variation of the negation detection depending on the entities recognized in the previous step, we explored the different NER approaches described bellow:

NER 1 : NegEx NER with dictionary list

We resorted to the strategy of NegEx to identify the entities, applying exact-matching to find in the document the words given by a list. This could be considered an approximation to the original NegEx strategy for Spanish given that the list consisted in a dictionary created with the medical entities (drugs and diseases) obtained from different sources.

The diseases of the list were the diseases and symptoms of the ICD-10. The drugs of the list were the drug families of the ATC classification, the drugs of Bot PLUS and the active ingredients of ICD-10.

NER 2 : NegEx NER with manual annotations list

We also resorted to the strategy of NegEx to identify the entities. However, in this case the list had the medical entities annotated by experts in the EHRs.

NER 3 : Conditional Random Fields (CRF)

The entities were recognized using the CRF algorithm (see Appendix B). For the creation of a basic CRF we transformed all the terms to lowercase and we used as features the prefixes and suffixes, which are the 4 first and last characters of the word. The reason is that some diseases are characterized by their affixes, for example, the suffix "*-tis*" indicates inflammation.

NER 4 : NegEx NER with manual annotations list + CRF

It consisted in the union of the entities detected by *NegEx NER with manual annotations list* and the entities detected by CRF.

NER 5 : Oracle

The recognized entities were those labeled by the experts. This could be considered as a perfect NER and shall provide us a upper threshold on the error propagation to the negation detection stage.

As explained before, to find the negation trigger words it is need a list to develop the exact-match. We used the list of negation trigger words created by (Costumero et al., 2014) for their adaptation of NegEx to Spanish. This list consists of the translation from English into Spanish of the 86 trigger words used in the original NegEx lexicon. We also included in the list 41 of the trigger words used in the Swedish NegEx version by (Skeppstedt, 2011). All in all, we accomplished a list of 121 different trigger words. Some words appeared in both list, for example, '*sin*' (meaning 'without') or '*ausencia de*' (meaning 'absence of'), that is, expressions that usually indicate the presence of a negation in any language.

Table A.1 shows the most frequent entities and negation trigger words in the same set of EHRs, respectively. The majority of the negation triggers

corresponded to 3 words (*'no'*, *'ni'*, *'sin'*). With only 3 words it was possible to detect approximately 98% of the negations.

	entity		frequency
1	*'hta'*	'ht'	40
2	*'fiebre'*	'fever'	25
3	*'disnea'*	'dyspnea'	24
4	*'dislipemia'*	'dyslipemia'	12
5	*'cardiomegalia'*	'cardiomegaly'	12
6	*'tos'*	'cough'	11
7	*'dolor*	'pain'	11
8	*'mareo'*	'sickness'	10
...	-	-	-

(a) Entities.

	negation trigger word		frequency
1	*'no'*	'no', 'not'	215
2	*'ni'*	"nor'	141
3	*'sin'*	'without'	83
4	*'niega'*	'denies'	3
5	*'ausencia de'*	'absence of'	3
6	*'ningún'*	'neither'	1
...	-	-	-

(b) Negation trigger words.

Table A.1: Lists of the most frequent entities and negation trigger words together with their frequency in the EHRs.

A.3.2 Evaluation

Given that the negation detection was developed with the entities recognized by five different NER approaches, we present first the results of the entity recognition and secondly the results of the negation detection. In both cases, we used the IxaMed-GS corpus (see Section 3.2.1) to infer and evaluate the models. It is important to explain that the evaluation was done with exact-match and partial-match (see Section 3.3).

Table A.2 shows the results achieved for the entity recognition. In this

case, we focused on the detection of diseases. We can see that the approach that offers worse results was *NegEx NER with dictionary list*. The main reason would be that the entities are not described by the experts in the same way that they appear in these dictionaries, then only few entities were found. We can also appreciate that using *NegEx NER with manual annotations list*, more entities were detected, but the precision was still low. Moreover, the results obtained with the *CRF* classifier outperformed those obtained with *NegEx NER with manual annotation list*. In general, with CRF the entities were detected with more precision. We also developed the evaluation of the approach that combines *NegEx NER with manual annotation list* and *CRF*. At first sight, the performance of this system should be better. However, *CRF* continued obtaining better results because *NegEx NER with manual annotation list* increased the number of FPs and, as a consequence, the precision worsened. Finally, as was expected, with the *Oracle* approach all the entities were found correctly given that this NER approach simply uses the entities labeled by the experts.

	Exact			Partial		
	P	R	F	P	R	F
NER 1	45.3	4.7	8.5	96.9	10.0	18.2
NER 2	36.5	39.5	37.9	69.4	74.1	71.7
NER 3	60.8	41.0	49.0	91.8	63.8	75.3
NER 4	36.4	44.7	40.1	70.2	83.3	76.2
NER 5	100.0	100.0	100.0	100.0	100.0	100.0

Table A.2: Precision (P), Recall (R) and F-measure (F) for the test set of the IxaMed-GS corpus in entity recognition with NegEx.

After having recognized the entities, the adapted NegEx had to detect the negation. Inherent to the cascade approach, the errors from the entity extraction are propagated to the negation detection. Note that, in the evaluation, a TP is an entity correctly identified as negated by the system.

Table A.3 shows the results achieved for the negation detection. According to this, the negation detection using the entities obtained with *NegEx NER with dictionary list* had the worst results (the f-measure for partial-match was 11.8). With this approach only few entities were detected, then, it was not possible to obtain a high recall for the negation detection. More-

over, the negation detection using the entities of *CRF* improved the results of the previous system (the f-measure for partial-match was 57.1). The idea was that the performance of NegEx would be better with these entities. However, the *NegEx NER with manual annotation list* approach was better than the previous one because it was able to recognize more diseases (the f-measure for partial-match was 73.8). To be precise, the recall was higher. Next, we evaluated the union of *NegEx NER with manual annotation list* and *CRF* for the entity recognition. This approach helped to improve the results obtained with *CRF* but not with *NegEx NER with manual annotation list* because the number of negated entities detected correctly increased less than the number of those detected incorrectly (the f-measure for partial-match was 64.7). Finally, we can see the performance of NegEx if all the entities labeled by experts would be recognized. With exact-match the results were better than the offered by the rest of approaches. Nevertheless, it did not happen for the partial-match given that, for example, the system could recognize two entities that experts labeled as one (the f-measure for partial-match was 66.4).

	Exact			Partial		
	P	R	F	P	R	F
NER 1	45.5	3.5	6.5	81.8	6.3	11.8
NER 2	42.1	35.9	38.8	80.2	68.3	73.8
NER 3	59.8	34.5	43.8	78.0	45.1	57.1
NER 4	39.2	35.9	37.5	67.7	62.0	64.7
NER 5	87.4	53.5	66.4	87.4	53.5	66.4

Table A.3: Precision (P), Recall (R) and F-measure (F) for the test set of the IxaMed-GS corpus in negation detection with NegEx.

A.4 Negation detection with CRF

The second approach used for the detection of negated medical entities was a machine learning system implemented with the CRF algorithm (Lafferty et al., 2001) (turn to Section B.3 for more information about CRF). The main reason was that machine learning methods were proven able to outperform regular expressions Cruz et al. (2010).

The creation of a supervised system for negation detection needs a big amount of annotated data to represent the information in a big-dimensional space, but this is not always possible to acquire. Above all, in a language different to English, which counts on a variety of tools and resources in comparison to others (Névéol et al., 2018) (see Section 1.2). Furthermore, in the medical domain, documents such as EHRs are subject to confidentiality agreements. For negation detection in the medical domain in Spanish, we only found publicly available the corpora created in (Cruz et al., 2017; Marimon et al., 2017). At the same time, in EHRs such as the used in this work, one word can be represented with different word-forms, using or not abbreviations, or with misspellings. This causes that, with a symbolic representation based on word-forms, one concept can have multiple representations. Turn to Section 3.2 to see more information about the EHRs. Both, the sparsity of data and the lexical variability tend to decrease the performance of the inferred models.

With the motivation of overcoming these issues, we decided to tackle negation detection by means of robust dense characterizations created with embeddings, following the idea presented in Chapter 5. Word-embeddings allow to represent words in a continuous space of small dimension and help inference algorithms to achieve better performance by grouping semantically related words (Mikolov et al., 2013a).

By contrast to mainstream techniques, that detect the negation cue and their scope, we were not interested in a broad scope, instead, we wish to mark absence of diseases or drugs. For this reason, we used an approach similar to NER, where the entities that we want to find are the negated ones.

A.4.1 Characterization

In order to train the model with the CRF classifier, it is necessary to represent the information through a set of representative features. The features used to detect the negated entities can be grouped in word-based representations ("Words"), embedding-based representation ("Embeddings"), clusters derived from the embeddings ("Cluster"), entity labels ("Entity") and negation trigger-word labels ("Trigger"). The features of each group are described below:

- **Words**
  - **Word-form**: Surface form of the tokens derived from FreeLing-Med analyzer (Oronoz et al., 2013). For example, the word-form of the disease '*dolor torácico*' (meaning 'chest pain') is 'dolor_torácico'.
- **Embeddings**
  - **uEHRs**: Vector corresponding to each word-form created with the in-domain dataset. For example, for the word '*medicación*' (meaning 'medication') the vector is '-2.300574, 0.68526, -2.457934, -1.15529, -2.444549'.
  - **SBWCE**: Vector corresponding to each word-form created with the out-domain dataset. For example, for the word '*medicación*' (meaning 'medication') the vector is '-0.332582, -6.167073, 3.17304, 0.504322, -0.667798'.
- **Clusters**:
  - **K-means**: Cluster assigned to each word using the k-means algorithm, which assigns the vector to the cluster with the nearest centroid. For example, for the word '*medicación*' (meaning 'medication') the cluster is '231'.
  - **Brown**: Cluster assigned to each word using the Brown algorithm, which merges those clusters for which the loss in the average mutual information is least. For example, for the word 'medicación' (medication) the cluster is '10011110111'.
  - **Brown truncated**: Aforementioned Brown cluster truncated to reduce the granularity. For example, for the word '*medicación*' (meaning 'medication') the cluster is '1001111011'.
- **Entity**
  - **Manual**: Label in BIO notation indicating if the token belongs (BI) or not (O) to a medical entity according to the annotations made by the experts. For example, for the word '*fiebre*' (meaning 'fever') the label is 'B-Grp_Enfermedad'.

  - **CRF**: Label in BIO notation indicating if the token belongs (BI) or not (O) to a medical entity according to the predictions made by a CRF classifier. The features used for this classifier were the word-form, lemma, POS and semantic tag (see Appendix B). For example, for the word '*antibiótico*' (meaning 'antibiotic') the label is 'B-Grp_Medicamento'.

- **Trigger**

  - **Label**: Label in BIO notation indicating if the token belongs (BI) or not (O) to a negation trigger word. For example, for the word '*no*' the label is 'B-Neg_TriggerWord'.

The embeddings of the in-domain corpus were created with GloVe (Pennington et al., 2014) and the embeddings of the out-domain corpus with skipNgram (Ling et al., 2015) (turn to Section 3.2.2 for further details of the unannotated corpora). Using in both cases a window of size 10 and yielding vectors of 300 components, as was done for the embeddings used in Chapter 5 and Chapter 6. The embedding-based characterizations the dimensions of the vectors were reduced to 5 component. This was done by means of the Principal Component Analysis (PCA), using the Weka libraries (Hall et al., 2009). Regarding the clusters, for the k-means cluster we used the algorithm implemented in word2vec (Mikolov et al., 2013a), for the Brown cluster we used the Brown algorithm (Brown et al., 1992) and the truncation was done using a maximum of 10 bits.

Combining the aforementioned features, we created different characterizations to represent the tokens of the EHRs. Firstly, we created the baselines. The first baseline (**B1**) makes use of word-forms. The second baseline (**B2**) exploits the embedding corresponding to the word-forms. In this way, we could observe if simple embedding-based features outperformed word-based features. Secondly, we included features related to the entities, the negation trigger word and the clusters. The characterization **C2.1** uses the word-embedding together with the CRFs entity label and negation trigger word label. Specifically, the characterization **C2.2** adds to the previous characterization the features corresponding to the clusters. These characterizations enabled us to compare again words and embeddings but, in this case, enhanced with features supposed useful for negation detection. The characterizations **U2.1** and **U2.2** are like the previous ones but using the gold entities

in order to have an upper threshold. Table A.4 summarizes the alternative characterizations together with the total number of features employed.

	Words	Embeddings		Clusters			Entity		Trigger	
Representation	Word-form	EHRs	SBWCE	K-means	Brown	Brown truncated	Manual	CRF	Label	Dimension
B1	✓									1
B2		✓	✓							10
C2.1		✓	✓					✓	✓	12
C2.2		✓	✓	✓	✓	✓		✓	✓	15
U2.1		✓	✓				✓		✓	12
U2.2		✓	✓	✓	✓	✓	✓		✓	15

Table A.4: Features used for each characterization employed to represent the documents. The last column shows the dimension of the feature-space.

A.4.2 Evaluation

The evaluation of the detection of negated entities was done using the IxaMed-GS corpus (see Section 3.2.1) and the hold-out evaluation scheme (see Section 3.3). Note that the precision, recall and f-measure were calculated at two levels: exact-match and partial-match (turn to Appendix B for more information about these evaluations). To infer the negation detection models we used a freely available implementation of CRF, CRF++ (Kudo, 2005). The templates were fine-tuned on the basis of the scores achieved on the development set. To be precise, the template chosen was a window [-2,-1,0,1,2] for the EHRs embeddings, [-3,-2,-1,0,1,2,3] for the SBWCE embeddings, [-2,-1,0,1,2] for the entity label, [-1,1] for the negation trigger word and [-1,0,1] for the clusters. With this template, a second system was trained making use of both train and dev sets and, next, it was evaluated on the test set. Finally, the results provided by the CRF classifier for each characterization (Table A.4) are shown in Table A.5.

	Exact			Partial		
	P	R	F	P	R	F
B1	37.2	23.2	28.6	88.3	59.7	71.2
B2	42.0	24.5	31.0	92.0	58.3	71.4
C2.1	43.2	27.2	33.3	91.6	61.7	73.7
C2.2	42.9	27.8	33.7	91.8	63.8	75.3
U2.1	51.7	41.1	45.8	93.3	82.4	87.5
U2.2	50.8	40.4	45.0	93.3	83.0	87.8

Table A.5: Precision (P), Recall (R) and F-measure (F) for the test set of the IxaMed-GS corpus in negation detection with the CRF classifier.

First, we compared the results obtained with word-based features (baseline **B1**) and embedding-based features (baseline **B2**). These results show that just replacing the symbolic characterization by a dense characterization improved the performance. The f-measure increased from 28.6 to 31.0 for exact-match and from 71.2 to 71.4 for partial-match. After testing that the word-embeddings could be useful in the characterization for the identification of negated entities, we explored the use of the entity and negation trigger labels for the embedding-based characterization (**C2.1**). We can see that this experiment outperforms both baselines. Furthermore, the incorporation of the clusters created from the embeddings (characterization **C2.2**) can help to improve the negation detection of the baseline (baseline **B2**) in the majority of the cases. The f-measure improved from 33.3 to 33.7 for exact-match and from 73.7 to 75.3 for partial-match. It could be because of the incorporation of different granularity levels in the representation.

Regarding the upper threshold, we can see that with the manual annotations the f-measure was always higher than with the CRF predictions, as was expected. Specifically, the f-measure increased from 33.7 to 45.0 for exact-match and from 75.3 to 87.8 for partial match. It happened because the f-measure for the entity recognition was 64.2 for the exact-match and 86.4 for the partial-match. In addition, we observed that if we do not take into account the discontinuous entities during the evaluation, the f-measure increased from 45.0 to 54.2 for the exact-match and worsened from 87.8 to 76.8 for the partial-match (using the upper threshold **U2.2**). This reflects that the discontinuous entities are only found partially. We also observed that this system can be able to detect entities that appear negated by means of a prefix, for instance, '*asintomático*' (maning 'asymptomatic').

A.5 Conclusions

We explored two different approaches to recognize medical entities in EHRs written in Spanish. Firstly, we adapted the NegEx rule-based system. Secondly, we created a machine learning system with the CRF classifier using dense features.

NegEx originally tackles the entity recognition by means of exact-match. Nevertheless, exact-match requires a big effort to keep the lists updated, particularly to deal with spontaneous EHRs. We made an adaptation that enables the use of any entity recognition technique able to generalize and, hence, recognize misspelled entities. According to the results, the entity recognition system that offered better results was the *CRF* classifier, which recognizes the entities with higher precision. However, the negation detection was better with the *NegEx NER with manual annotation list* because this approach commits less errors among the detected negations than the union of both. We learned that we could extend NegEx to detect the negation implicit to several prefixes. We also observed that the presence of discontinuous entities makes the detection of negated entities challenging because sometimes the negation trigger word is not in the span considered by NegEx.

Detecting the negation with the CRF classifier, we observed that the use of embeddings instead of symbolic features was helpful to detect negated entities. Above all, in cases with sparse data, as happens with the EHRs. The best results were obtained with the characterization C2.2, that includes the clusters. In this case, the system was be able to detect entities negated with prefixes and the majority of the discontinuous entities were found partially. To the best of our knowledge, this is the first time that the negation detection was done using embedding in the characterization for texts written in Spanish.

Comparing both systems according to the upper threshold, we can see that for the exact-match evaluation the f-measure is higher with NegEx, but for the partial-match evaluation the f-measure is higher with CRF.

Medical Entity Recognition

B.1 Introduction

NER is an essential first step in extracting information from texts (Grishman, 2003). The main reason is that the information obtained with NER can be introduced later in other tasks such as relation extraction (Wang et al., 2018a).

As a result of the importance of NER, we can found different shared tasks such as *CoNLL-2002* (Tjong Kim Sang, 2002), *CoNLL-2003* (Tjong Kim Sang and De Meulder, 2003) and *GermEval 2014* (Benikova et al., 2003). *CoNLL-2002 shared task* and *CoNLL-2003 shared task* were devoted to language-independent named entity recognition. The first one was tested on texts written in Spanish and Dutch and the second one was tested on texts written in English and German. *GermEval 2014 shared task* was devoted to named entity recognition for German.

NER can be defined as the task of identifying and semantically classifying named entities in text (Patrick and Wang, 2005). In the medical domain, medical entities such as diseases or drugs are found and this is called MER. In this work we develop MER, where we can distinguish two parts: i) medical entity boundary identification and ii) medical entity classification. MER entails some difficulties with respect to the classical NER (Ben Abacha and Zweigenbaum, 2011). On the one hand, the medical entities have a high terminological variations (different terms express the same concept). On the other hand, the evolution of entity naming (new names and abbreviations).

The different approaches used to develop MER can be divided in: i)

dictionary-based methods, ii) rule-based methods, and iii) machine learning methods (Wang et al., 2018a). In this work, the methodology proposed for this task is based on machine learning, consists in the use of algorithms that perform the classification task. Specifically, we employ the CRF classifier. In this way, we perform both parts of MER simultaneously.

The rest of the appendix is organized as follows: Section B.2 presents the state-of-the-art about MER. Section B.3 explains the methods used for the recognition of the medical entities. Section B.4 shows the results obtained during the experimentation. Finally, Section B.5 is devoted to give a brief conclusion about this task.

B.2 Related work

In the approaches used in related works to develop the medical entity recognition, we can distinguish different supervised machine learning algorithms and also different types of features for the characterization.

Among the classifiers used for this task, we can find ME, SVM and CRF. For example, Lin et al. (2004) employed the ME classifier and added a post-process based on rules to improve the detection, creating and hybrid approach. Kazama et al. (2002) used SVM to develop the biomedical named entity recognition. The authors also compared SVM with ME concluding that SVM achieved better results. Settles (2004) and Tang et al. (2015) decided to employ the CRF classifier for this task. Moreover, to expedite the training speed, Tang et al. (2015) implemented the model training process on a parallel optimization program framework based on MapReduce. Ben Abacha and Zweigenbaum (2011) compared SVM and CRF classifiers and concluded that CRF yielded better results.

In the aforementioned works, different features for the characterization were explored. Among the features used for MER we can find orthographical features related with the use of capital letter or the presence of numbers in the token (Lin et al., 2004; Tang et al., 2015; Ben Abacha and Zweigenbaum, 2011). Morphological features that consist of the suffixes and prefixes of the words (using different character lengths) (Lin et al., 2004; Kazama et al., 2002; Settles, 2004; Tang et al., 2015; Ben Abacha and Zweigenbaum, 2011). POS features (Lin et al., 2004; Kazama et al., 2002; Tang et al., 2015) and lemma features (Ben Abacha and Zweigenbaum, 2011). We can also find semantic features indicating the semantic category of the word (Tang et al.,

2015; Ben Abacha and Zweigenbaum, 2011) and trigger word and keyword features (Tang et al., 2015).

In these works, the representation is based on word-based features. However, we can also find works such as (Copara et al., 2016), where the authors used dense characterizations for NER. Copara et al. (2016) tried 4 different representations: 1) Brown clustering, 2) Clustering embeddings, 3) Binarized embeddings, and 4) Distributional prototypes. These were created with the Spanish Billion Corpus and the English Wikipedia, that is to say, they use cross-lingual word representation. The experiments were conducted with the CRF classifier and the authors concluded that the cluster-based features improved the baseline whereas the embedding-based features worsened it.

Regarding the corpus, Lin et al. (2004) and Kazama et al. (2002) used the GENIA corpus. This corpus forms part of the corpus of the BioNLP/NLPBA 2004 shared task used by Settles (2004) and Tang et al. (2015). Specifically, the training set is the GENIA corpus, which consist of 2,000 abstracts from Medline database, and the test set consists of 404 abstracts also from Medline database. Ben Abacha and Zweigenbaum (2011) employed the i2b2/VA 2010 challenge corpus, formed by clinical texts. Finally, Copara et al. (2016) turned to the CONLL 2002 corpus, which is written in Spanish and is not related with the biomedical domain.

B.3 Entity recognition with CRF

For MER we used the CRF classifier (Lafferty et al., 2001), given that the majority of the tools created for NER are based on CRF (Wang et al., 2018a). This classifier involves a probabilistic framework for labeling and segmenting sequential data. CRF constructs a conditional model $p(Y|X)$ to create a discriminative framework from the jointly distributed variables X and Y, instead of modeling the marginal $p(X)$. X are observation sequences and Y their corresponding label sequences. That is to say, it takes into account the information of the earlier and later tokens to make the predictions.

We characterized each token in two different ways denoted as i) basic features and ii) complex features. The basic features consist in information about the morphology of the words. These are the lowercase word-form, prefixes and suffixes. The complex features involve information about the syntax an the semantics of the words. These are the lemma, the POS and the semantic tag, which were obtained with the FreeLing-Med analyzer (Oronoz

et al., 2013). The entities were tagged using the BIO format: B (beginning), I (inside), O (outside).

B.4 Evaluation

The MER task was evaluated using the IxaMed-GS corpus (see Section 3.2.1) and the hold-out evaluation scheme (see Section 3.3). Note that the precision, recall and f-measure were calculated at two levels with the software of the SemEval task "Analysis of Clinical Text" (Nakov and Zesch, 2014):

- **Exact-match**: The entity found by the system is the same as the entity annotated by the experts. The comparison is made using the offsets, that is to say, the position of the first and last characters of the entity in the text.
- **Partial-match**: The entity found by the system and the entity of the manual annotation overlap, that is to say, the initial offset of one of the entities is between the offsets of the other entity.

First, the evaluation was done using the train set for training and the dev set for evaluating. Finally, we give the final results training with the train and dev sets and evaluating with the test sets. To infer the MER models we used a freely available implementation of CRF, CRF++ (Kudo, 2005). The template chosen for the basic features was a window [-2,-1,0,1,2] for the word-form, [-2,-1,0,1,2] for the prefixes and [-2,-1,0,1,2] for the suffixes. The template chosen for the complex features was a window [-1,0,1] for the word-form, [-2,-1,0,1,2] for the lemma, [-1,0,1] for the POS and [-2,-1,0,1,2] for the semantic tag. The results for the "Grp_Enfermedad" and "Grp_Medicamento" entity types are shown in Table B.1. In this table we can also see the evaluation of FreeLing-Med for NER (Oronoz et al., 2013), which is used as baseline since the complex features were created with information obtained from it.

In these results we can observe that the MER task performed better with the CRF classifier that uses the features based on FreeLing-Med (complex features). Furthermore, the results obtained with partial-match were better that those obtained with exact-match (with a difference of approximately 20.0). This could happen due to the terms with more than one word and the discontinuous entities. With the terms that comprise more than one word,

the scope of the recognized entity can be smaller or bigger than the given in the manual annotations. With the discontinuous entities, in some cases the model only detects one of the parts of this entity.

Classifier	Features	Exact			Partial		
		P	R	F	P	R	F
FreeLing-Med	-	46.5	45.0	45.7	75.6	71.5	73.5
CRF	basic	70.7	45.2	55.1	94.3	61.6	74.5
CRF	complex	75.1	56.5	64.5	96.3	73.2	83.2

Table B.1: Precision (P), Recall (R) and F-measure (F) for the test set of the IxaMed-GS corpus for MER.

In addition, we also show the results achieved for MER with CRF and the complex features using the IxaMed-E corpus. The reason is the experiments with automatic entities made in Chapter 7. Specifically, we used the entities recognized by CRF to create the drug-disease pairs and see their influence on the performance of ADR extraction (see Section 7.3). Given that these experiments were evaluated for the dev and the test set, in Table B.2 we show the results obtained for MER in both sets.

In comparison with the results obtained for the test set of the IxaMed-GS corpus, we can observe that there was only a slightly improvement for the partial-match. The f-measure changed from 83.2 to 84.8.

Classifier	Features	Exact			Partial		
		P	R	F	P	R	F
CRF	complex	64.4	51.7	57.4	90.7	79.8	84.9

(a) Model inferred with the train set and evaluated with the dev set.

Classifier	Features	Exact			Partial		
		P	R	F	P	R	F
CRF	complex	63.6	52.1	57.2	89.6	80.4	84.8

(b) Model inferred with the train and dev sets and evaluated with the test set.

Table B.2: Precision (P), Recall (R) and F-measure (F) for MER using the CRF classifier with the IxaMed-E corpus.

B.5 Conclusions

We tackled in a shallow manner the MER task using CRF as classifier. The best performance was obtained using the syntactic and semantic features, that is, the lemma, the POS and the semantic tag. Although FreeLing-Med offered worse results for this task, its information was beneficial as features for the CRF classifier.

C

Detailed results: Adverse Drug Reaction detection with dense representations and Random Forest

In this appendix we show the detailed results of the experiments developed for ADR detection using dense representations and the RF classifier in Chapter 5. Given that these experiments yielded 40 results with different metrics, they were represented in Figure 5.4 for compactness. Now, apart from showing these results in Figure C.1 grouped by the approach and corpus employed to generate the embeddings, we tabulate the detailed results. Tables C.1, C.2, C.3 and C.4 contain the precision, recall and f-measure for the positive ($\oplus$) and the negative ($\ominus$) class of the experiments developed training with the train set and evaluating with the dev set.

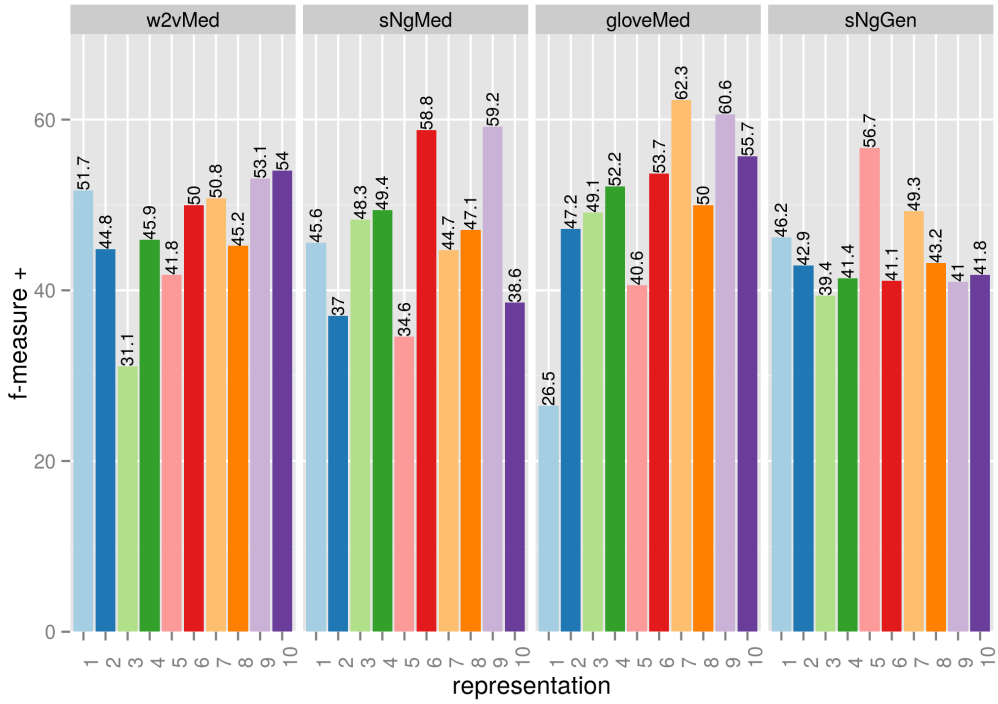


Figure C.1: F-measure of the positive class with the 10 representations presented in Table 5.2 for the dev set of the IxaMed-GS corpus using the Random Forest classifier. The embeddings were extracted using three different techniques (denoted as w2v, sNg, and glove to refer to word2vec, skipNgram, and GloVe respectively) and from two sources, denoted by the suffix, where Med stands for in-domain medical source and the suffix Gen stands for the general out-domain source.

Representation	Precision	Recall	F-measure	Class
1	53.6	50.0	51.7	$\oplus$
	89.0	90.3	89.6	$\ominus$
2	46.4	43.3	44.8	$\oplus$
	87.5	88.8	88.1	$\ominus$
3	46.7	23.3	31.1	$\oplus$
	84.6	94.0	89.0	$\ominus$
4	45.2	46.7	45.9	$\oplus$
	88.0	87.3	87.6	$\ominus$
5	37.8	46.7	41.8	$\oplus$
	87.4	82.8	85.1	$\ominus$
6	50.0	50.0	50.0	$\oplus$
	88.8	88.8	88.8	$\ominus$
7	48.5	53.3	50.8	$\oplus$
	89.3	87.3	88.3	$\ominus$
8	43.8	46.7	45.2	$\oplus$
	87.9	86.6	87.2	$\ominus$
9	50.0	56.7	53.1	$\oplus$
	90.0	87.3	88.6	$\ominus$
10	51.5	56.7	54.0	$\oplus$
	90.1	88.1	89.1	$\ominus$

Table C.1: Results of the 10 representations of Table 5.2 with **word2vec** embeddings and **in-domain** corpus for the dev set of the IxaMed-GS corpus using the Random Forest classifier.

Representation	Precision	Recall	F-measure	Class
1	48.1	43.3	45.6	⊕
	87.6	89.6	88.6	⊖
2	41.7	33.3	37.0	⊕
	85.7	89.6	87.6	⊖
3	50.0	46.7	48.3	⊕
	88.2	89.6	88.9	⊖
4	40.4	63.3	49.4	⊕
	90.6	79.1	84.5	⊖
5	40.9	30.0	34.6	⊕
	85.2	90.3	87.7	⊖
6	52.6	66.7	58.8	⊕
	92.1	86.6	89.2	⊖
7	37.0	56.7	44.7	⊕
	89.0	78.4	83.3	⊖
8	57.1	40.0	47.1	⊕
	87.4	93.3	90.3	⊖
9	51.2	70.0	59.2	⊕
	92.7	85.1	88.7	⊖
10	40.7	36.7	38.6	⊕
	86.1	88.1	87.1	⊖

Table C.2: Results of the 10 representations of Table 5.2 with **skipNgram** embeddings and **in-domain** corpus for the dev set of the IxaMed-GS corpus using the Random Forest classifier.

Representation	Precision	Recall	F-measure	Class
1	23.7	30.0	26.5	$\oplus$
	83.3	78.4	80.8	$\ominus$
2	40.5	56.7	47.2	$\oplus$
	89.3	81.3	85.2	$\ominus$
3	56.5	43.3	49.1	$\oplus$
	87.9	92.5	90.2	$\ominus$
4	46.2	60.0	52.2	$\oplus$
	90.4	84.3	87.3	$\ominus$
5	38.2	43.3	40.6	$\oplus$
	86.9	84.3	85.6	$\ominus$
6	48.6	60.0	53.7	$\oplus$
	90.6	85.8	88.1	$\ominus$
7	61.3	63.3	62.3	$\oplus$
	91.7	91.0	91.4	$\ominus$
8	47.1	53.3	50.0	$\oplus$
	89.2	86.6	87.9	$\ominus$
9	55.6	66.7	60.6	$\oplus$
	92.2	88.1	90.1	$\ominus$
10	54.8	56.7	55.7	$\oplus$
	90.2	89.6	89.9	$\ominus$

Table C.3: Results of the 10 representations of Table 5.2 with **GloVe** embeddings and **in-domain** corpus for the dev set of the IxaMed-GS corpus using the Random Forest classifier.

Representation	Precision	Recall	F-measure	Class
1	54.5	40.0	46.2	⊕
	87.3	92.5	89.9	⊖
2	33.3	60.0	42.9	⊕
	89.1	73.1	80.3	⊖
3	34.1	46.7	39.4	⊕
	87.0	79.9	83.3	⊖
4	42.9	40.0	41.4	⊕
	86.8	88.1	87.4	⊖
5	51.4	63.3	56.7	⊕
	91.3	86.6	88.9	⊖
6	34.9	50.0	41.1	⊕
	87.6	79.1	83.1	⊖
7	41.9	60.0	49.3	⊕
	90.1	81.3	85.5	⊖
8	36.4	53.3	43.2	⊕
	88.3	79.1	83.5	⊖
9	32.1	56.7	41.0	⊕
	88.3	73.1	80.0	⊖
10	37.8	46.7	41.8	⊕
	87.4	82.8	85.1	⊖

Table C.4: Results of the 10 representations of Table 5.2 with **skipNgram** embeddings and **out-domain** corpus for the dev set of the IxaMed-GS corpus using the Random Forest classifier.