

Borja Antonio Espejo García

Site-Specific Rules Extraction in Precision Agriculture

Departamento
Informática e Ingeniería de Sistemas

Director/es
Zarazaga Soria, Francisco Javier
López Pellicer, Francisco Javier

<http://zaguan.unizar.es/collection/Tesis>



Reconocimiento – NoComercial – SinObraDerivada (by-nc-nd): No se permite un uso comercial de la obra original ni la generación de obras derivadas.

© Universidad de Zaragoza
Servicio de Publicaciones

ISSN 2254-7606



Universidad
Zaragoza

Tesis Doctoral

**SITE-SPECIFIC RULES EXTRACTION IN
PRECISION AGRICULTURE**

Autor

Borja Antonio Espejo García

Director/es

Zarazaga Soria, Francisco Javier
López Pellicer, Francisco Javier

UNIVERSIDAD DE ZARAGOZA
Informática e Ingeniería de Sistemas

2019

Site-Specific Rules Extraction in Precision Agriculture

Borja A. Espejo-García

May 2019

PhD Advisors:

Dr. F. Javier Zarazaga Soria

Dr. Francisco J. López Pellicer



Departamento de
Informática e
Ingeniería de Sistemas
Universidad Zaragoza

Acknowledgements

I would like to express my sincere gratitude to my PhD. advisors Prof. F. Javier Zarazaga and Prof. Francisco J. López for the support of my PhD research, for their patience and motivation. Their guidance helped me in all the time of research and writing of this thesis.

My sincere thanks also goes to the rest of *Advanced Information Systems Laboratory* (IAAA) partners (and family in some “geek” way): Dr. Javier Lacasta, Dr. Javier Nogueras, Dr. Rubén Béjar, Dr. M. A. Latre, for their encouragement and insightful comments. Of course, it would be impossible to forget the initial Pedro Muro’s push.

I would also like to thank the professors that take me in during the research internships: Prof. Manuel Pérez-Ruiz, Prof. Manuel Fernández Delgado, and Prof. Spyros Fountas, for offering me opportunities in their research groups and leading me working on diverse exciting projects.

I thank my countless labmates: Alejandro, Dayany, Jorge, Sergio, Víctor, Yaneisy, Yinet, Enrique, Jorge (Seville), Nikos, George, Loukas, Matina, Nikoleta, Sofia, Mihalis, Vangelakis, Dinos, Katerina, Hristos and a large etcétera, for the stimulating (and funny) discussions during the coffee and lunch breaks. Without you, the thesis would have around 100 more pages; but it would be less interesting.

Also I thank my refusès friends in University of Zaragoza: Alba, Jean de l’Osè, and Dr. Manuel Bedia for striking each other with the fire pokers when a neoliberal sophism was com-

mitted. Of course, I don't forget Regli and Uxua for remind me that life is not just work.

Last but not the least, I would like to thank my family: my parents Antonio and Pilar, for supporting me day by day without mistake (this thesis is also yours); and my uncles and cousins for keeping an eye on their youngest nerd cousin.

Resumen Ejecutivo

El incremento sostenible en la producción alimentaria para satisfacer las necesidades de una población mundial en aumento es un verdadero reto cuando tenemos en cuenta el impacto constante de plagas y enfermedades en los cultivos. Debido a las importantes pérdidas económicas que se producen, el uso de tratamientos químicos es demasiado alto; causando contaminación del medio ambiente y resistencia a distintos tratamientos. En este contexto, la comunidad agrícola divisa la aplicación de tratamientos más específicos para cada lugar, así como la validación automática con la conformidad legal. Sin embargo, la especificación de estos tratamientos se encuentra en regulaciones expresadas en lenguaje natural. Por este motivo, traducir regulaciones a una representación procesable por máquinas está tomando cada vez más importancia en la agricultura de precisión.

Actualmente, los requisitos para traducir las regulaciones en reglas formales están lejos de ser cumplidos; y con el rápido desarrollo de la ciencia agrícola, la verificación manual de la conformidad legal se torna inabordable.

En esta tesis, el objetivo es construir y evaluar un sistema de extracción de reglas para destilar de manera efectiva la información relevante de las regulaciones y transformar las reglas de lenguaje natural a un formato estructurado que pueda ser procesado por máquinas. Para ello, hemos separado la extracción de reglas en dos pasos. El primero es construir una ontología del dominio; un modelo para describir los desórdenes que producen las enfermedades en los cultivos y sus tratamientos. El

II

segundo paso es extraer información para poblar la ontología. Puesto que usamos técnicas de aprendizaje automático, implementamos la metodología MATTER para realizar el proceso de anotación de regulaciones. Una vez creado el corpus, construimos un clasificador de categorías de reglas que discierne entre obligaciones y prohibiciones; y un sistema para la extracción de restricciones en reglas, que reconoce información relevante para retener el isomorfismo con la regulación original. Para estos componentes, empleamos, entre otras técnicas de aprendizaje profundo, redes neuronales convolucionales y “long short-term memory”. Además, utilizamos como baselines algoritmos más tradicionales como “support-vector machines” y “random forests”.

Como resultado, presentamos la ontología PCT-O, que ha sido alineada con otras ontologías como NCBI, PubChem, ChEBI y Wikipedia. El modelo puede ser utilizado para la identificación de desórdenes, el análisis de conflictos entre tratamientos y la comparación entre legislaciones de distintos países. Con respecto a los sistemas de extracción, evaluamos empíricamente el comportamiento con distintas métricas, pero la métrica F_1 es utilizada para seleccionar los mejores sistemas. En el caso del clasificador de categorías de reglas, el mejor sistema obtiene un macro F_1 de 92,77% y un F_1 binario de 85,71%. Este sistema usa una red “bidirectional long short-term memory” con “word embeddings” como entrada. En relación al extractor de restricciones de reglas, el mejor sistema obtiene un micro F_1 de 88,3%. Este extractor utiliza como entrada una combinación de “character embeddings” junto a “word embeddings” y una red neuronal “bidirectional long short-term memory”.

Executive Summary

The sustainable increase of food production to feed an on the rise world population is a real challenge when we take into account the threat of crop diseases on agricultural production. Since crop diseases can produce important environmental and economic losses, the use of chemical treatments is too high and causes pollution and resistance to different treatments. Within this context, farming community envision the achievement of site-specific chemical treatment applications with its respective automatic compliance checking. However, treatment application specifications are found in regulations expressed with natural language. For this reason, translating regulations into a machine-processable representation is becoming increasingly important in plant protection management.

Currently, the requisites to translate human-oriented regulations into formal rules are far from being accomplished; and with the rapid development of agricultural science, manual compliance checking of farming practices will be unapproachable. In this thesis, the objective is to build and evaluate a rules extraction system in order to effectively distill the essential information from regulations and transform the natural language rules into a structured format that could be processed by a machine.

Although there are some possible solutions for rule extraction, there is no evidence on the suitability of these works in precision agriculture domain. In this thesis, we have separated the rule extraction development in two steps. The first step is to construct a domain ontology; a model to describe the outbreaks

IV

that pests produce to crops and the approved ways to treat them. The second step is to extract information in order to populate the ontology. Since we use a machine learning approach, we implement the MATTER methodology to develop the whole annotation task in the rule extraction development. Afterwards, we train a rule category classifier that discerns between obligations and prohibitions; and a rule constraints extractor that recognizes target information and labels them with pre-defined information tags based on the ontology. Target information represents the meaning of the rules and retain the isomorphism with the original regulations. For the machine learning-based components, we employ, among other deep learning techniques, convolutional and long-short-term memory networks. Moreover, we use as baselines more traditional algorithms such as support vector machines and random forests.

As results, we present the PCT-Ontology, which has been aligned with other ontologies such as NCBI, PubChem, ChEBI and Wikipedia. The model can be used for tasks such as the identification of outbreaks, analysis of site-specific related conflicts with the treatments, and comparison of solutions among country legislations. Regarding the extraction systems, we empirically evaluate the performance with different metrics, but F_1 score is used to select the best systems. In the case of the rule category classifier, the best system obtains a macro F_1 of 92.77% and a binary F_1 of 85.71%. This system uses a bidirectional long short-term memory with word embeddings as input. Regarding, the rule constraints extraction, the best system achieves a micro F_1 of 88.30%. This extractor is a combination of character and word embeddings and a bidirectional long short-term memory.

Contents

1	Introduction	1
1.1	Background	1
1.1.1	Agriculture and Integrated Pest Management	1
1.1.2	Precision Agriculture	3
1.2	Motivation	8
1.3	Research Hypothesis	12
1.4	Scope	13
1.5	Contributions	23
1.6	Organisation of this thesis	24
2	Related Work	27
2.1	Feature Engineering	28
2.1.1	Rule-based techniques	29
2.1.2	Unsupervised techniques	30
2.2	Machine learning	32
2.2.1	Shallow Models	33
2.2.2	Deep neural network models	35
2.3	Rule Extraction	38
3	PCT Ontology	45
3.1	Introduction	45
3.2	Ontology Design Development	50
3.2.1	Structure of the PCT-O	50
3.2.2	Ontology construction	55

3.3	Results	61
3.4	Discussion	63
3.5	Summary	66
4	Rule Category Classification	69
4.1	Introduction	69
4.2	Rule Classifier Development	73
4.2.1	Preprocessing	74
4.2.2	Machine Learning Training	78
4.3	Experiments	81
4.3.1	Local structure of agricultural standards in Spain	81
4.3.2	Gold Corpus Creation	83
4.3.3	Preprocessing	84
4.3.4	Rule Classifier Evaluation	84
4.4	Results	86
4.4.1	Shallow Machine Learning Results	87
4.4.2	Deep models Results	95
4.5	Discussion	97
4.6	Summary	99
5	Rule Constraints Extraction	101
5.1	Introduction	101
5.2	Deep NN Architectures for IE	107
5.3	PCT-O Extension	112
5.4	Experiments	113
5.4.1	Constraints in Spanish agricultural stand- ards	114
5.4.2	Preprocessing	119

5.4.3	Hyperparameter tuning	119
5.4.4	Evaluation	121
5.5	Results	122
5.5.1	Word-Level evaluation	123
5.5.2	Rule-level evaluation	126
5.6	Discussion	127
5.7	Summary	129
6	Conclusions	131
6.1	Summary of Contributions	131
6.2	Future Work	134
6.2.1	Revision	134
6.2.2	New approaches	141
6.3	Final Conclusion	144

List of Tables

4.1	Skip-Gram Algorithm Parameters.	76
4.2	Original Embedding Corpus Statistics.	76
4.3	Corpus Statistics.	84
4.4	Parameter specification for the algorithms.	88
4.5	Summary of the algorithms with best prohibition precision.	89
4.6	Summary of the algorithms with best prohibition recall.	91
4.7	Summary of the algorithms with best prohibition F_1 score.	92
4.8	Selected LSTM Hyperparameters after cross val- idation.	96
4.9	Summary of the deep networks architectures with best binary and macro F_1 score.	97
5.1	Label types used for annotation.	116
5.2	Gold Corpus Statistics.	118
5.3	Number of Labels in the corpus.	118
5.4	Hyperparameters	121
5.5	Nomenclature used to describe neural architec- tures in table 5.6.	123
5.6	Architecture's F_1 score per label type.	124
5.7	Comparison of the IE systems at rule-level.	127

List of Figures

1.1	Example of real time site-specific treatment application.	5
1.2	Site-Specific regulations as a key enabler of a real time site-specific application.	8
1.3	Rule Extraction workflow. This thesis is focused on the part inside the box.	15
1.4	The MATTER methodology step to address the development of an Rule Extraction system (Pustejovsky and Stubbs, 2013).	16
2.1	Structure of a deep RNN classifying “Apply until flowering”.	36
2.2	Structure of a deep CNN classifying “Do not apply after fruiting”.	38
3.1	PCT-O development: Contextualization of this chapter in the whole workflow.	46
3.2	Plant affections and their treatment ontology.	54
3.3	Ontology population process.	59
3.4	Classification of pests.	61
4.1	Rule Category Classification: Contextualization of this chapter in the whole rule extraction workflow.	70
4.2	Traditional rule classification pipeline.	73

4.3	Deep rule classification pipeline.	80
4.4	Example of document from in the Spanish official registry of phytosanitary products.	83
4.5	F ₁ score comparison of the different NLP techniques used.	93
4.6	F ₁ score comparison of the different NLP techniques used.	94
4.7	F ₁ score comparison of the different machine learning algorithms evaluated.	95
5.1	Rule Constraints Extraction: Contextualization of this chapter in the whole rule extraction workflow.	102
5.2	Neural sequence labelling architecture for sentence “Apply until flowering”.	108
5.3	Extension of PCT-O in order to represent the complexity of a chemical treatment application.	113
5.4	Example of labeled rules.	118
5.5	Comparison of layers performance in deep models.	126
6.1	Rule Extraction workflow parts to be refined and improved in future work.	135
6.2	Rule Extraction workflow parts to be faced in future work.	141

Nomenclature

ACC	Automatic Compliance Checking
AI	Artificial Intelligence
BLSTM	Bidirectional Long-Short Term Memory
CBOW	Convolutional Neural Network
CNN	Convolutional Neural Network
CRF	Conditional Random Fields
DL	Deep Learning
FMIS	Farm Management Information System
IAA	Inter-Annotator Agreement
IE	Information Extraction
LSTM	Long Short Term Memory
ML	Machine Learning
NLP	Natural Language Processing
NLU	Natural Language Understanding
NN	Neural Networks
PA	Precision Agriculture

PPP	Plant Protection Product
RF	Random Forest
RNN	Recurrent Neural Network
SL	Supervised Learning
SVM	Support Vector Machines
XML	eXtensible Markup Language

Chapter 1

Introduction

A problem well put is half solved.

John Dewey
The Pattern of Inquiry

1.1 Background

1.1.1 Agriculture and Integrated Pest Management

Food production must increase by 70% in order to feed a world population that is expected to reach 9.6 billion by 2050 (Foley, 2011). This challenge becomes trickier when we take into account the scarcity of new arable land, the effects of climate change on agricultural production and the societal demand for producing healthy food while reducing environmental impact (Garrett et al., 2006; Blanc and Reilly, 2017). Within this context, the inclusion of new technologies and methods in agriculture, arises as one of main solutions to overcome these chal-

lenges. Moreover, these solutions must take into account that effective agricultural production is really complex because it is affected by various heterogeneous factors like environmental conditions, soil characteristics (Benjamin and Gallic, 2018), water availability (Agacayak and Keyman, 2018), harvesting practice (Jain et al., 2016) and crop diseases (Nazir et al., 2018).

Crop diseases are one of the major threats because they can produce substantial social, environmental and economic losses. For example, early blight represents one of the most common diseases in the world and can cause a significant decreasing of yields and many lesions in fruits (Blancard, 2012). It is calculated that crop yield losses caused by weeds are about 32% while those caused by pests and pathogens are 18% and 15% respectively (Oerke, 2006). Therefore, identifying crop disorders and establishing a treatment according to its severity is a critical topic that has been studied through the years. For instance, in the area of crop protection, several approaches, such as Integrated Pest Management (IPM), are presented as ways to develop more sustainable crop protection strategies, such as early disease detection and disease-specific chemical applications (Schut et al., 2014). Within IPM, planning, compliance checking and documentation are becoming more relevant in order to implement control measures to improve the suppression of diseases. Traditionally, these tasks have been managed by the farmers own expertise; however, this approach, as for any activity carried out by humans, is subject to psychological and cognitive phenomena that may lead to bias, optical illusions and, finally, to error (Bock et al., 2010).

The use of chemical treatments, which are cheap and ef-

fective compared to mechanical procedures, is too high and has caused waste of chemicals, ground environmental pollution and resistance to different treatments (Qi et al., 2009). This situation has led to a growing governmental pressure on farming practices in order to limit the usage of chemicals. This pressure comes in the form of a complicated regulatory framework to shift managerial tasks to a new paradigm that requires more attention on environmental impact and terms of delivery (Sigrimis et al., 2000; Dalgaard et al., 2006). For example, subsidies are often an incentive for the farmer to engage in a sustainable production in order to reduce the application of different agrochemical inputs such as pesticides and fertilisers. Under these conditions, farmers need to combine a huge amount of data, besides intelligent machinery, to achieve crop management efficacy, while adhering to governmental regulations. For this reason, farming is hosting a fourth revolution triggered by use of Information and Communication Technology (ICT) through different approaches such as Precision Agriculture (PA).

1.1.2 Precision Agriculture

PA is the scientific domain that deals with the management of spatial and temporal variability through ICT (Blackmore et al., 2003). It optimises farm management tasks, such as the control of diseases, while reducing the ecological footprint and consequently boosting consumer acceptance. In the context of PA, the requirement to spray Plant Protection Products (PPP) as homogeneously as possible all over the field is out of date. Site-specific application of PPPs takes into consideration the spatial

variability avoiding environmental impacts on non-target areas and mitigates the emission of greenhouse gases (Corwin and Plant, 2005). Several works have shown the benefits of using chemical applications supported by PA technologies. For example, by using detailed information about present weed plant species, their growth stages and plant densities in a field, the herbicide consumption can be reduced by 40% on average (Jørgensen et al., 2007). In Wu et al., 2009, by using herbicide spraying according to weed density and using half dosage chemical herbicides, they could produce the same effect as the whole dosage in a low density weed area.

These advances are highly related to the improvement of information management through Farm Management Information System (FMIS). FMISs, which evolved from computerised record-keeping systems, refer to planned systems for collecting, processing, storing and disseminating data in the form needed to carry out a farm's operations and functions (Sørensen et al., 2010). With this technology, the different groups of stakeholders involved in the agricultural activities (e.g., farmers and policy actors) can manage many different and heterogeneous sources of information (e.g., meteorological web services, regulations, guidelines, etc.) that need to be integrated in order to provide economically and environmentally accurate decisions, which include (among others) development of sustainable agriculture, disorders detection, crop recollection timing and pricing, and the adherence to regulations.

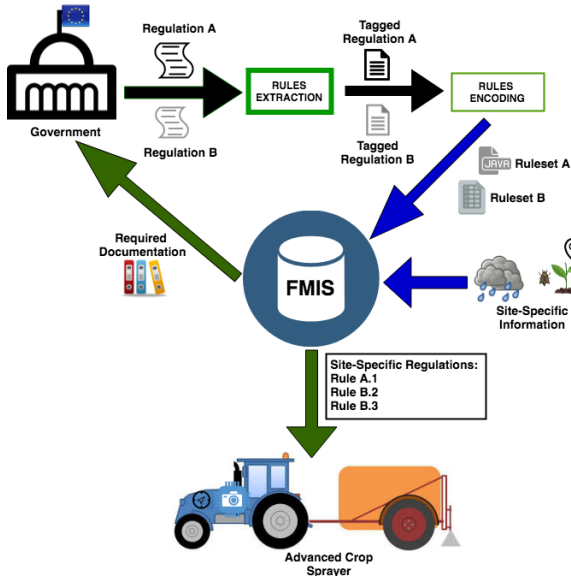


Figure 1.1: Example of real time site-specific treatment application.

An example of the site-specific treatment application envisioned in this thesis is shown in Figure 1.1. All the presented components will be detailed in the following paragraphs. The aim of this system is to do a real-time diagnosis beside treatment prescription and application based on site-specific information and regulations. As it can be observed, different crops are regulated by different regulations depending on their location

(e.g., near body waters), the country where they are going to be sold (different countries have different regulations), etc. It is important to note that there would be another approaches such as generating a prescription map that would be fed into the computer system of the machinery. A more schematic representation of the necessities to achieve site-specific treatment application is shown in Figure 1.2.

In this site-specific application example, we can observe some challenges that imply three requirements that have to be satisfied from a scientific and technical point of view:

1. Site-specific information: There is a need to monitor enough site-specific information about pest and disease status of the field crops which should be treated with PPPs. Advances in crop monitoring, such as positioning systems and sensors, allows farmers to acquire vast amount of site-specific data which ultimately can be used to enhance decision making (Fountas et al., 2006). One of these sensors are RGB and hyper-spectral cameras which in combination with automated nondestructive computer vision techniques can play a valuable role in detecting and recognizing crop diseases by using leaves images when human assessment is unsuitable, unreliable or unavailable (Mohanty et al., 2016; Tang et al., 2017). Currently, the conjunction of Deep Learning (DL), which is a subfield of Machine Learning (ML), and transfer learning, together with the development of Graphics Processing Units (GPUs) has provided a powerful tool for recognition and classification of crop diseases (Ferentinos, 2018). In the case

of large-scale cultivations, this system could be attached to autonomous agricultural vehicles or unmanned aerial vehicles with autonomous flight control, to accurately and timely locate phytopathological problems throughout the cultivation field, using continuous image capturing and sending them to an FMIS besides other relevant data (field coordinates, crop plant, available PPPs, etc.).

2. **Advanced crop sprayers:** It is necessary a field crop sprayer being able to apply through direct injection different kind of PPPs independently from each other at the same time. Within the last few years, technical development of field crop sprayers have reached a high level saving PPP, reducing drift and enhancing the operator protection (Wegen, 2017). Just like the monitoring machinery explained above, these sprayers can be integrated with current agricultural vehicles, which are able to execute autonomous or remote actions at all levels of agricultural production, such as mechanical weeding, application of fertiliser, or harvesting of fruits (Zhang and Pierce, 2013). Moreover, its integration with further information systems containing site-specific information such as the growth stage, the specific weather conditions and previous applications registry will contribute to lower the risks, save PPPs and increase the economical benefit of plant protection measures.
3. **Site-Specific Regulations:** It is necessary that all treatment regulations that orchestrate how sprayers perform

their operations and which site-specific information is necessary, are loaded within the FMIS. Different FMIS components, such as Rule Extractors or Rule Encoders, should have access to the different data repositories, which are accessible at local, national and European level. Finally, FMIS should also have the possibility to send the field documentation to the government, so that the compliance can be officially checked.

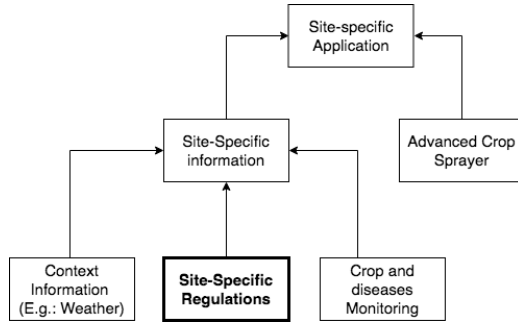


Figure 1.2: Site-Specific regulations as a key enabler of a real time site-specific application.

1.2 Motivation

Due to the increasing importance of regulations, and with the objective of site-specific applications of PPPs, one of the goals of

PA is Automatic Compliance Checking (ACC). In several countries, national plant protection plans are incorporated in external open data repositories that contain information on cross-domain national and international production standards in the form of legal regulations or quality assurance labels; in addition to all types of production guidelines for farm activities. Each farm must take into account all the relevant regulations, which may vary from field-to-field, and ensure that they are respected during operations in order to make economical and environmentally-sound decisions. This process is burdensome for farmers because many steps are manual. Consequently, farmers often experience an overload of information (Sørensen et al., 2010).

Literature has already shown that manual compliance checking is time consuming, costly and error-prone (Zhang, 2015; Dragoni et al., 2016). The main problem to achieve ACC is that in even advanced digital societies, the regulations published by governments and the standards publishers describing what is permitted or forbidden, are usually expressed only in Natural Language (NL); and it is very unlikely to have a machine-processable representation of the conditions contained in such documents. For this reason, incorporating regulations and standards directly into FMIS is becoming increasingly valuable in order to reduce time, cost and errors. Once these regulations are incorporated within the software, there is a large potential for farmers to be supported, for instance, with site-specific plant protection regulations by applying automated logical reasoning techniques to infer whether a violation has occurred. This can lead to more specific and better treatment

recommendations; significantly easing the administrative burden on farmers imposed by complying with the wide range of regulations which they must consider to receive subsidies and avoid fines.

Despite the numerous approaches tackling the problem of automatically moving from a NL legal text to the respective set of machine-readable conditions, results are still unsatisfiable and incorporation of regulations in information systems remains a major open challenge that still requires too manual effort (Zhang, 2015; Dragoni et al., 2016). One reason is that extracting rules from the NL regulatory text and transforming them into the executable rules requires both domain knowledge and the satisfaction of several technological requirements.

In the PA domain, Nash et al., 2011 provided a conceptual framework to approach the ACC problem by analyzing rule extraction. According to them, it is foreseeable that in the next years up to 90% of the agricultural production rules will be automatically extracted because these five prerequisites will be technically feasible:

1. Categorisation of each rule as an obligation, prohibition or documentation.
2. Possibility of formal modelling.
3. Automated machine interpretation.
4. The objectivity of the required assessment.
5. The availability of the required data.

Currently, these prerequisites are far from being accomplished and an example of the regulations heterogeneity can be found in the data collections provided by the Spanish Ministry of Agriculture ¹ where the description of how to control each type of pest is distributed among multiple heterogeneous textual sources. For example, each document has a layout slightly different from the rest and the names of the pests in the document title are variants of those used in the pest description. This lack of interoperability affects critically tasks requiring some degree of data integration such as identifying the different crops affected by a single organism, finding similitude in the treatment of different species, and comparing the approved pesticides in different countries. Additionally, as new products and techniques are frequently approved, a continuous review is required (Labussi ere et al., 2010). This happens not only in Spain, but also in many other countries such as United Kingdom ², United States ³ or Canada ⁴.

We can conclude that automatic rule extraction is an interesting and useful challenge to be addressed. Once this rule extraction system has been developed, we foresee the following potential benefits in farming:

1. To allow earlier identification of potential non-compliance chemical applications.

¹<http://www.mapama.gob.es/>

²<https://secure.pesticides.gov.uk/pestreg/>

³<https://www.epa.gov/pesticide-registration>.

⁴<https://www.canada.ca/en/health-canada/services/consumer-product-safety.html>

2. To promote the adoption of FMIS. A future possibility could be the development of an automated pesticide prescription system that orchestrate all the necessary information to allow the purchase of appropriate pesticides by the farmers.
3. To explore what-if scenarios; since a farmer could experiment with different protection strategies and check their compliance in a more time-efficient manner.
4. To avoid violations of regulations and consequently protect the environment.

1.3 Research Hypothesis

In general, the regulations about pest control are published in heterogeneous and human-oriented formats, so intensive manual labour is required to identify the most suitable solution for a given pest or disease. Moreover, with the rapid development of agricultural science and technology, the complexity and specificity of regulations is going to be increased in the next years. Consequently, manual compliance checking of farming practices against various regulations will be unapproachable. In this thesis, the research question is as follows:

“Is it possible to build a Rule Extraction Engine for pesticide regulations with current modelling and learning techniques in the field of artificial intelligence?”

From the research hypothesis come the following research questions:

1. Is it feasible to create and populate a conceptual model where the relevant concepts of crop treatment application can be considered?
2. Is it possible for a machine learning system to detect the subtle linguistic difference between prohibitions and obligations?
3. Are there any suitable deep learning architectures in order to automatically extract meaningful information from the regulations?

1.4 Scope

The research work has the following scope:

- **Rule Extraction:** In this thesis, we start from the framework proposed in Nash et al., 2011; and we adapt the MATTER methodology (explained in the next point) to develop an approach to automatically extract requirements or rules from different regulatory texts to obtain a logic representation for further automated reasoning within the ACC context. Although different authors have proposed different methods to address the challenge, our construction of a rule extraction system can be divided into several tasks that altogether constitute the rule extraction pipeline demonstrated in Figure 1.3.

In this pipeline, the first step is the construction of an agricultural conceptual model; secondly the extraction of modelled information from regulatory documents; and thirdly the information transformation into normalised rules. This third part is out of the scope of this work and will be explored in the future. Regarding the extraction of information (step 2), our approach involves developing a set of algorithms and combining them into the pipeline: (1) ML algorithms for rule category classification and (2) ML algorithms for rule constraints extraction. The rule category classifier discerns between obligations and prohibitions in a regulatory text corpus. This step is critical because an error implies that the meaning of the rule is inverted. For example, a rule such as “*Do not apply in crops with fruits destined to preserve*” could be interpreted as “*Apply in crops with fruits destined to preserve*”. On the other hand, the rule constraints extractor, which is a kind of Information Extraction (IE) system, recognizes the phrases that carry target information and labels them with pre-defined information tags based on the conceptual model from step 1 (e.g.: maximum number of applications, site of the application, feasible phenological stages, etc.). Target information is needed to check a specific type of regulatory requirement or constraint; they represent the meaning of the rules and retain the isomorphism with the original text.

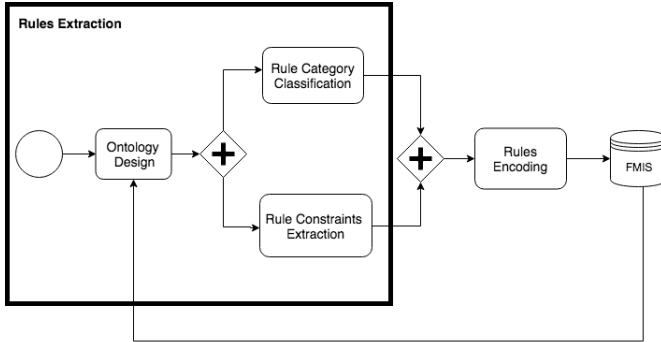


Figure 1.3: Rule Extraction workflow. This thesis is focused on the part inside the box.

- Annotation Methodology:** In order to extract information from rules or any other text source, it is necessary to previously annotate which information is relevant or meaningful for a specific task or goal. Semantic annotation is an important part of using computers for processing natural languages that has been applied in different tasks such as Medical Record Processing and Sentiment Analysis (Wang et al., 2018). In this work, we have used the MATTER methodology to develop our rule extractor project (Pustejovsky and Stubbs, 2013). This framework outlines six major stages to automatic annotation, corresponding to each letter in the word: Model, Annotate, Test, Train, Evaluate and Revise. Figure 1.4 provides a

visualization of the framework. These steps describe a general methodology for creating annotated corpora and ML tasks of all different types, from part-of-speech tagging to discourse analysis. Since the MATTER cycle is a general description of the process, there is no conflict between it and the existing annotation standards. Due to the uncertain nature of the research, we leverage the flexibility of the MATTER cycle because it is agnostic to the decisions made regarding corpus selection, annotation tools or representation formats.

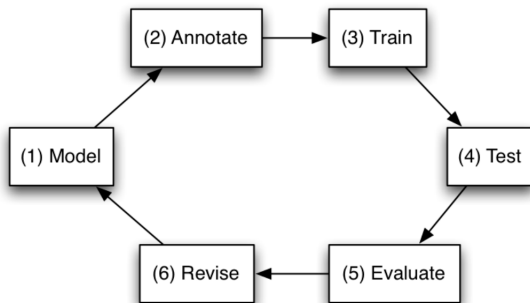


Figure 1.4: The MATTER methodology step to address the development of an Rule Extraction system (Pustejovsky and Stubbs, 2013).

A prerequisite to start with the MATTER cycle, is to define the particular goal of the annotation task as a guide

to determine method for collecting a representative corpus. Naturally, as the MATTER cycle progresses it is possible that the goal may be redefined. In our research, the goal will be to populate a model that can recommend pesticides extracting rules from regulations.

- **Knowledge Representation:** It can be observed that the first step in MATTER methodology is the modelling of the phenomena (i.e., domain knowledge representation) that is going to be annotated. This model represents the relevant or meaningful information according to a specific goal. The ability to recognise this domain concepts in text is one of the requirements in rule extraction. In order to represent the domain knowledge, a semantic model in combination with Natural Language Processing (NLP) techniques can be used to automatically formalise knowledge in free text capturing the meaning of a domain in a structured manner by distilling its more relevant concepts. Ontology is a widely-used type of semantic model; it is defined as “an explicit specification of a conceptualization” (Gruber T, 1995). The model represents both the information that is going to be collected from the corpus during the course of the annotation, and the information that will be extracted later using ML or other NLP systems. A semantic model helps to recognise the semantic information tags of each extracted information instance.

To create the ontology, it is rewarding to search for background work to understand existing theories of the phenomena or creating or adopting an existent ontology of the

phenomenon. In the case of agriculture, a number of ontology development efforts are under way. Some of the main ontologies in the domain of weed control and crop protection are the Global Agricultural Concept Scheme (GACS) (Baker et al., 2016), which combines AGROVOC (Stelato, 2002), the CAB Thesaurus and the NAL Thesaurus into one ontology, the Plant Ontology (Jaiswal et al., 2005) and Crop Ontology (Skofic et al., 2012). Unfortunately, required concepts for crop protection such as treatment and outbreak are not yet included in these ontologies, which limits their immediate usefulness. For this reason, we create our own ontology based on agricultural standards, existing ontologies and expert knowledge.

- **Corpus Selection:** Although there are different regulatory texts, we have evaluated the research hypothesis with regulations obtained from the Spanish official registry of phytosanitary products. It is important to note that Spanish is the second most widely-spoken language on Earth. Moreover, Spanish agriculture has a valuable role inside the EU (European Commission, 2018; European Commission, 2018). In the Spanish phytosanitary authorizations, regulations can be structured into a set of individual rules which roughly fulfil with the requirements to achieve an automatic compliance assessment (Nash et al., 2011). (i) The analysed phytosanitary rules are obligations or prohibitions which could be encoded in a machine-readable form. The source does not contain documentation rules. (ii) The rules have a recurrent terminology, which could

be modelled with an ontology. (iii) They also have a discrete outcome and, finally, (iv) the required data inputs could be available in future years with the use of technologies related to PA such as remote sensing, computer vision and FMIS. It is significant to remark that this approach could also be adaptable to other types of regulations in the farming domain.

- **Annotation Scheme:** Once the conceptual model is created, we must annotate the selected corpus to create the gold corpus where ML algorithms will learn. The output of this process is the annotated regulations text. Of course, the annotated corpus needs to be consistent to train/test an algorithm. In this work, we adopt a two-level annotation. We do a shallow annotation for the sentence level to discern between obligations and prohibitions. For the IE part we use an standard annotation scheme. These schemes provide an encoding where each information tag indicates a certain type of meaning related to the conceptual model. There are different schemes in the literature such as IOB1, IO and IOE2 (Krishnan and Ganapathy, 2005); but in this work we only use the IOB2 because it is widely used and is more fruitful if the information instances of the same kind are adjacent since it enables to locate the boundaries (Yano, 2018).
- **Natural Language Processing:** Once the gold corpus is created, the step prior to train a ML model is to generate feature representations that are fed into ML algorithms to perform TC or IE. These features are built through

pipelines that often involves various NLP modules to extract different linguistic characteristics. The determination of which NLP modules and resources are used and which features should be extracted is called feature engineering (Sharnagat, 2014). A large portion of the previous research effort in IE has been spent on developing effective feature sets for different subtasks such as named entity recognition, relation extraction, event extraction etc.

However, despite much effort on hand-designing feature representations, the resulting feature sets might be not necessarily optimal. Moreover, feature engineering is very time-consuming and very often yields incomplete non-satisfactory sets. For this reason, in this work, we contrast the traditional feature engineering methods against two additional approaches for word representation: word embedding, which is trained from a large amount of text, and character-based representation, which can capture orthographic features of words through neural networks.

- **Machine Learning Training:** It is the process by which a ML algorithm is taught to recognize which features have more weight to create the desired output from the system. ML provides a powerful framework to assimilate data; for this reason, the advantage of our approach regarding traditional rule-based approaches, is that with the advances in the field of ML and the creation of new datasets, we expect to improve the performance of our system and extract more accurate information. Semi-supervised learning, distant supervision and unsupervised learning are beyond the

scope of this work that focuses on supervised learning. As explained in the rule extraction point, we use a ML approach in two parts of the process:

- **Rule Category Classification:** Text classification is one of the principal tasks of ML and it refers to the process of designing proper algorithms to enable computers to assign weights to linguistic features from sentences and classify them automatically in predefined classes (in this thesis: prohibition and obligation). In the 1960s, people began to study text classification through manual classification rules according to language phenomena and grammars; but by the 1990s, people began to study ML based automatic classification technology. This method is first trained by annotating data and learning discrimination rules. In this research, we use this latter approach through different ML algorithms. We have also done some preliminary experiments with deep learning for text classification. Experimenting with this approach has sense because, firstly, large amounts of data to train a deep model are generated every day. And secondly, the power of computing provided by GPUs makes possible the training of deep models. In the current thesis, we employ Convolutional Neural Networks (CNN) and Long-short Term Memory (LSTM), which constitute specific classes of deep models that have been applied to various agricultural and food production challenges

(Mohanty et al., 2016; Amara et al., 2017).

- **Rule Constraints Extraction:** Entities and events are central objects of languages. It is therefore crucial for computers to recognize such objects so that they can come closer to the understanding of human languages. To locate and classify entity mentions in text into predefined classes to populate the model is essentially the target of IE. In our work, the task is to develop a system capable of automatically detecting any mentions of crop treatments and their attributes (site, dosage, frequency, intervals duration) as well as mentions of phenological stages and other relevant treatment restrictions. As in the case of rule category classification, we also use DL through CNN and LSTM layers.
- **Machine Learning Evaluation:** Once the ML algorithms have been trained on the corpus and tuned on the evaluation data, it is necessary to run them on the test data for measuring their performance. Given the complexity of ML methods, they resist formal analysis methods; and systematic experimentation is a key part of applied ML (Langley, 1988). The objects of our experiment are programs and we must learn empirically about its behaviour on our specific problems through controlled experiments. In these experiments, all known independent variables are held constant and modified one at a time in order to determine their impact on the dependent variable; in our case, the rule extraction performance.

In this research, the performance is evaluated in terms of precision, recall, and F_1 score metrics. Their specific equations will be presented in the following chapters. It is essential to note that, both 100% recall and precision are desired. However, given the inherent trade-off between the two measures, it is challenging to accomplish that goal. Therefore, the ultimate goal for our ACC system is to achieve the highest F_1 score. If the scores are not good enough, the algorithm and the features must be changed, and the new output is evaluated again until a satisfactory level of performance is reached.

The Revision state of the MATTER cycle is the point at which the entire project, from corpus selection to ML evaluation results, is reviewed. Topics for revision include: aspects of the task that may have contributed to poor performance, changes to the task that could result in improved performance later on, and new applications of the task that could be done successfully in the future, such as expanding the task to a new language or ontology model enrichment with new concepts.

1.5 Contributions

This thesis aims to develop a framework to automatically extract information from NL regulations. Starting from this objective, the main contributions of the research are the following:

1. This thesis presents a study of the current techniques to

translate NL regulations into a set of formal rules.

2. This thesis develops its own methodology to translate regulations into rules by adapting the MATTER methodology and the framework presented in Nash et al., 2011.
3. This thesis presents a domain ontology developed to support decision in crop treatment application. The use of ontologies is a classical solution to deal with heterogeneity and interoperability problems.
4. This thesis evaluates the combination of different ML algorithms, NLP techniques and resampling methods for classifying texts rules between prohibitions and obligations.
5. This thesis presents a comparison between deep learning and “traditional” machine learnings algorithms in order to classify rules as prohibitions or obligations.
6. This thesis presents a linguistic-agnostic end-to-end sequence labeler in order to automatically label meaningful information from regulations.

1.6 Organisation of this thesis

Chapter 2 presents a survey of the related literature with the main concepts, technologies and topics managed along this thesis. We present essential background concepts related to ML and

NLP. The rest of the chapters include their own state-of-the-art in order to provide more specific contexts.

Chapter 3 introduces the creation of a domain ontology (PCT-O) for rule extraction. Moreover, another popular biological and chemical ontologies are studied in order to reuse and extend some of their concepts. A preliminary IE extraction is also faced in this chapter. Additionally, this chapter open some modelling deficiencies inside PCT-O that are faced in the next chapters.

Chapters 4 develops a rule classifier through different ML models. This chapter presents a systematic empirical study using different ML algorithms, NLP preprocessing techniques and resampling methods in order to find the best combination able to discern between prohibitions and obligations. We also present a preliminary comparison between traditional ML algorithms and DL algorithms on the same task.

Chapter 5 evaluates different end-to-end DL models for IE and demonstrates their applications on rule constraints extraction (available phenological stages for treatment, maximum number of applications, etc.). Moreover, due to the cyclic incremental development process that MATTER provides, an extension of the PCT-O is also provided. This model extension gives a support for the labels tagged by the IE system.

Chapter 6 presents a discussion about the limitations of the developed rule extraction system, and points to some new research lines that could be interesting to extend our work.

Chapter 2

Related Work

Why should there be the method of science? There is not just one way to build a house, or even to grow tomatoes. We should not expect something as motley as the growth of knowledge to be strapped to one methodology.

*Ian Hacking
Representing and
Intervening*

In this chapter, we provide some relevant background knowledge, necessary to put the rest of this thesis in context. Most of the following sections are devoted to give an up-to-date view on the landscape of rule extraction. But first of all, we go over the most important pillars that are needed to understand the process of rule extraction.

2.1 Feature Engineering

Feature engineering is the process of using domain knowledge of the data to create features that make ML algorithms working. Feature engineering is a fundamental step in our workflow but it is both tricky and expensive because of its informal nature. As (Ng et al., 2013) states, “Applied ML is basically feature engineering”. Feature engineering is composed of different techniques focused on dealing with missing data, transforming categorical values, dealing with skewed distributions and outliers or non-scaled variables. In the world of natural language, there is a subset of techniques that can be included under the NLP umbrella.

NLP is the attempt to extract a fuller meaning representation from free unstructured text. In order to deal with language complexities, it is necessary to extract linguistic features that help to understand accurately the phenomena. The hypothesis that has been driving NLP is the one set by Chomsky, 1957: “The fundamental aim in the linguistic analysis of a language L is to separate the grammatical sequences which are the sentences of L from the ungrammatical sequences which are not sentences of L and to study the structure of the grammatical sequences.”

To achieve this goal, NLP typically makes use of linguistic features such as part-of-speech (noun, verb, adjective, etc.) and grammatical structure (either represented as phrases like noun phrase or prepositional phrase, or dependency relations like subject-of or object-of). In many cases, NLP relies on ML to derive

meaning from human languages by analysis of the text semantics and syntax. Mainly, we can find two approaches: Rule-based and unsupervised techniques.

2.1.1 Rule-based techniques

Rule-based (or dictionary-based) feature engineering in NLP makes use of various techniques to extract linguistic features. For example, semantic representations, such as a lexicon of words, grammatical properties through a set of grammar rules and often other resources thesaurus of synonyms or abbreviations. Another supervised features may be orthographic features, Parts Of Speech (POS) and morphological characteristics (e.g., prefixes and suffixes) among others. In this thesis, we use the following techniques:

1. POS tagging: it is the process of marking a word in a text as corresponding to a particular part of speech based on both its definition and its context (Brill, 1992).
2. Stemming: it consists of removing any attached suffixes and prefixes from words because singular and plural forms of a noun or different verb forms are semantically the same in many contexts. By removing them, it is reduced the redundancy and complexity in the model. In Luo et al., 2015, they use this technique along with DL.
3. N-grams: this technique attempts to solve the problem of information loss when transforming a document into a set of independent words because sometimes word context

matters. Single tokens are known as unigrams, and pairs of tokens are known as bigrams. For instance, in Ur Rahman et al., 2016 they use unigrams and bigrams to extract diseases from medical texts.

4. Stop words removal: functional words and punctuation are removed by default in rule category classifier. These steps remove words that are not relevant such as some articles (e.g., “the” and “a”), and pronouns. It is important to highlight that there is no single universal list of stop words, and they depend on the context. We use part of the NLTK stop words list ¹.

2.1.2 Unsupervised techniques

Besides using rule-based techniques to represent individual words, currently it is possible to take advantage of very large unlabelled text data to learn word features to enrich models obtained from a small gold corpus. Many methods have been proposed to induce unsupervised word representations. Mainly, they can be classified into two categories: clustering-based word representations and distributed word representations. Due to the success of the latter approach in recent literature (e.g.: Habibi et al., 2017), we use it in this thesis.

Distributed word representations, also called word embeddings, generate a low-dimensional, real-valued and dense vector for each word using neural language models. It is a more informative way of representing words compared to NLP one-hot

¹<https://bit.ly/2H7tjwf>

encodings where the representation of all the words is independent of each other. Word embeddings representations can capture latent semantic or grammatical information of words because it allows words with comparable meaning to have close mathematical representation based on which words are used in similar contexts. For example, they can capture semantic (dis)similarities between tokens that are not visible from their morphological surface (e.g. ‘Flowering’ and ‘Fruiting’). This approach is highly related to the transfer learning technique in computer vision. Transfer learning is the idea of overcoming the isolated learning paradigm and utilizing knowledge acquired for one task to solve related ones. It is important to note that there are cases when transfer learning can lead to a drop in performance. Negative transfer refers to scenarios where the transfer of knowledge from the source to the target does not lead to any improvement, but rather causes a drop in the overall performance of the target task.

Word embeddings are based on the works of Bengio et al., 2003 and Collobert et al., 2011. Bengio et al., 2003 proposed a neural network architecture to predict the next word given the previous ones; Collobert et al., 2011 proposed a neural network architecture that checks whether a text fragment is valid. Subsequent studies were mainly based on these architectures. Mikolov et al., 2011 simplified previous architectures, and presented two novel models: continuous bag-of-words (CBOW) model and Skip-gram with negative sampling with much lower computational cost. As reported in Mikolov et al., 2013, the Skip-gram is slightly superior to the CBOW model and is the current state-of-the-art word-embedding method. Moreover, Skip-gram is also

very efficient to train, works in an online fashion, and scales well to huge corpora (billions of words) as well as very large word and context vocabularies (Balikas and Amini, 2016).

Although embeddings with Skip-gram are a significant step forward compared to bag-of-words and have potential of improving the performances of ML based systems, their usefulness is typically determined by the problem domain (Wolpert and Macready, 1995); and thus, it is compelling to understand the background of these models and corpora. Different variants typically differ in the corpus they originate from, such as Wikipedia, news articles, etc., and the differences in the embedding models.

2.2 Machine learning

ML is the name given to the area of Artificial Intelligence concerned with the development of algorithms that learn or improve their performance from experience or previous encounters with data. They are said to learn (or generate) a function that maps particular input data to the desired output. Within ML, there is a subfield called Supervised Learning (SL) that will be evaluated in this thesis. SL is the search for algorithms that reason from externally supplied instances to produce general hypotheses, which then make predictions about future instances (Kotsiantis, 2006). More formally, given a set of K predefined classes Y ($|Y| = K$) and an object X , we need to choose a class $y \in Y$ that captures the nature of X . For instance, in text classification, the predefined classes Y involve the semantic rule categories (i.e, “prohibition”, “obligation”, etc) while the objects

correspond to the words inside the sentences.

Since the appearance of DL in the classification world, it is needed to make a distinction between shallow models and deep neural network models.

2.2.1 Shallow Models

Contrary to deep models (see Section 2.2.2), which are based on neural networks; until 5 years ago, ML models were based on other paradigms in which the neural model was not the most relevant one. Currently, these models could be characterised for being less complex (shallower) than deep neural networks. When it comes to ML, there is no free lunch (Wolpert and Macready, 1995). We must test all possible algorithms for data at hand to identify the best suitable algorithm. Besides of picking the right algorithm we must also choose the right configuration of the algorithm for a dataset by tuning the hyper-parameters. Furthermore, there are several other considerations for choosing the winning algorithm such as computational complexity, explainability, and ease of implementation. In this thesis, we use:

1. Logistic regression: it arises from the desire to model the posterior probabilities of classes (in this work, obligation and prohibition) via linear functions in the feature space (in this work, the words after feature engineering) while ensuring that the probabilities sum to one and remain in the range of $[0,1]$ (Kleinbaum and Klein, 1994; Friedman et al., 2008).

2. Naive Bayes: this classifier (Langley and John, 1995), is based on the popular Bayes probability theorem. It is known for creating simple yet effective linear models. For example, this approach yielded excellent results when applied to spam classification and disease prediction (Saad et al., 2012). The main difference between naive Bayes and logistic regression is that the former optimises the joint probability and the latter optimises the posterior probability.
3. Support Vector Machines (SVM): until the rising of DL, they provided state-of-the-art text classification models because of their robustness to high dimensionality problems (Cortes and Vapnik, 1995). An SVM model treats examples (in this work, the rules after feature engineering) as points in space, and these points are mapped so that the examples of different categories are separated by a gap that is as wide as possible. This model is also a representation of examples as points in space that are mapped as described above; however, contrary to logistic regression, the gaps between classes of points are as wide as possible. Because of the excellent results that SVM algorithms have achieved in a wide variety of domains, including in the agricultural field (Zhou et al., 2014), they have rapidly gained popularity.
4. Random forest (RF) methods: they use decision trees (i.e., a forest) with random independently sampled vectors, and all trees in the forest have the same distribution (Breiman, 2001). They are popular algorithms in the ML community,

and they have been used recently in the agricultural field (e.g., Brillante et al., 2015; Görgens et al., 2015).

2.2.2 Deep neural network models

Since Krizhevsky et al., 2012, deep models have arisen as the state-of-the-art solution in many artificial intelligence fields. At the lowest level, DL involves computing a function that maps some inputs to their corresponding outputs. Although the function itself is just a bunch of addition and multiplication operations passed through a non linear functions; by stacking a batch of these layers, functions are universal learners that can learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in fields where enough of data is accessible such as speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics (Lecun et al., 2015). Deep neural networks exploit the property that many natural signals are compositional hierarchies, in which higher-level features are obtained by composing lower-level ones. In NL, hierarchies exist from sounds to phones, phonemes, syllables, words and sentences. In other words, DL's major advantage is the capacity to automatically induce effective feature representations (automatic feature engineering) from noisy data. Two neural networks architectures have taken importance during the past years: Recurrent Neural Network (RNN) and Convolutional Neural Networks (CNN).

2.2.2.1 Recurrent Neural Networks

Recurrent Neural Network (RNN) models have been shown to be promising techniques due to their ability to learn from the context surrounding the words in a sequence (Lipton et al., 2015). Currently, the mainstream approach is to consider a sentence as a sequence of tokens (characters or words) processed in sequential order, from left to right and using a deep model to “memorise” the whole sequence in its internal states. Since RNNs suffer from vanishing and exploding gradient problems, a subtype of RNNs called Long Short-Term Memories (LSTMs) (Hochreiter and Schmidhuber, 1997) have arisen and they are good at modelling varying length sequential data achieving state-of-the-art results for many problems in NLP, such as neural machine translation, question answering and text classification (Liu et al., 2017; Lample et al., 2016; Ma and Hovy, 2016).

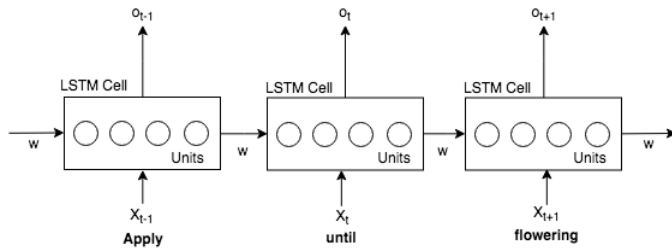


Figure 2.1: Structure of a deep RNN classifying “Apply until flowering”.

Although, LSTMs suffer from weakness of not utilising the future contextual information, bidirectional LSTM (BLSTM) (Schuster and Paliwal, 1997) addresses this issue by using two independent LSTMs (forward and backward) in which one processes the input sequence in the forward direction, while the other processes the input sequence in the reverse direction. A schematic structure of BLSTM is shown in Figure 2.1.

2.2.2.2 Convolutional Neural Networks

Although originally invented for computer vision, CNN models have subsequently been shown to be effective for NLP and have achieved excellent results (Collobert et al., 2011). Contrary to the classical artificial neural networks, whose training requirements are exhaustive, CNN use weight sharing which allow massive parallelization. There are four key ideas behind CNNs that take advantage of the properties of natural signals: local connections, shared weights, pooling and the use of many layers. At a base level, the weights of a CNN consist of filters that are convoluted (slide and multiply) through the provided signal.

Basically the training of a CNN involves, finding of the right values on each of the filters so that an input signal when passed through the multiple layers, activates certain neurons of the last layer so as to predict the correct class.

A schematic structure of a CNN next to a embedding representation is shown in Figure 2.2. It is also depicted the decision part of the architecture with a fully-connected neural network to obtain a final prediction. The convolutional layers act as feature

extractors from the input text whose dimensionality is then reduced by the pooling layers (not shown in the Figure 2.2), while the fully connected layers exploit the high-level features learned.

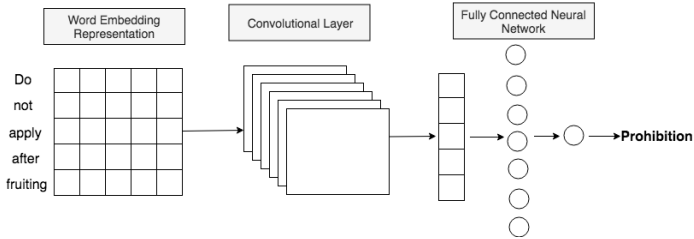


Figure 2.2: Structure of a deep CNN classifying “Do not apply after fruiting”.

2.3 Rule Extraction

Automatic rule extraction from NL text have a long history; and heterogeneous approaches have been proposed in the literature. For example, Engers et al., 2004 developed and implemented a knowledge based framework to extract concepts and norms by using linguistic techniques. Their desire was to reduce the knowledge acquisition bottleneck understanding the semantics of normative expressions in legal texts. However, they did not consider the identification of deontic modalities in rules, as in this thesis; and no evaluation of their automated norms extraction framework was provided; thus, results cannot be used as a

guide.

In Soria et al., 2005, authors addressed the problem of automatically enriching legal texts with semantic annotation of Italian legal texts exploiting NLP techniques: they tried to classify law paragraphs according to their regulatory content and to extract text fragments corresponding to specific semantic roles relevant for the regulatory paragraph. They envisioned this process as an essential prerequisite to effective indexing and retrieval of legal documents. The approach used in this work is quite similar to the ours; they created the SALEM tool to automatically tag the semantic structure of Italian law paragraphs through an integration of NLP and IE technology. Similarly to them, we also use this two-step approach to extract rules; however, the main difference is that, instead of ML, they used an incremental composition of shallow parsing with higher levels of syntactic and semantic analysis, leading to simultaneous, effective combination of low- and high-level text features for fine-grained content analysis.

Biagioli et al., 2005 faced the problem in an analogous way: (i) rule classification and (ii) IE of meaningful parts; but instead, they proposed a ML based framework for the semantic annotation of provisions to ease the retrieval process of norms. They used SVM to classify the rules. In our case, we extend this approach with a broader evaluation of ML algorithms besides SVM. Moreover, we use them within a combination of resampling and NLP techniques and additionally we evaluate each combination in a systematic way.

Kiyavitskaya et al., 2008 proposed a methodology for extracting stakeholder requirements, called rights and obligations,

from regulations where texts are annotated to identify fragments describing normative concepts, and then a semantic model is constructed from these annotations and transformed into a set of requirements. They presented the results from two empirical evaluations of a tool that extracts a conceptual model from regulatory texts. Again, ML does not appear as relevant in this research making an important difference with our research.

Francesconi, 2010 used ML and NLP techniques for extracting legal rules on the basis of a semantic model for legislative texts, which is oriented to knowledge reusability and sharing. Moreover the identified entities of the regulated domain can be a starting point to a bottom-up implementation of domain ontologies. A similarity with our work is that, instead of using pattern matching methods relying on lexico-syntactic patterns, they proposed ML techniques based on SVM. A slight difference is that they evaluated different weighting techniques inside the feature engineering step while in the case of this thesis, it is an experimental constant that is not evaluated.

Otherwise, Maat and Winkels, 2010 used ML for Dutch regulations. After an analysis of twenty Dutch laws, fourteen categories were defined and 88 different classification patterns (phrases) were designed to recognise them. These patterns were used to classify the sentences in eighteen other Dutch laws. Out of 592 sentences, 91% was classified correctly. Contrary to this thesis, they used a knowledge engineering approach. On the other hand, this study is similar to other related work in that classification is performed at sentence level (Francesconi, 2010; Biagioli et al., 2005).

Wyner et al., 2011 presented a linguistic-based approach to

extract deontic rules from regulations. In order to annotate texts, they used a tool called GATE ². They discussed a pilot study in which they used C&C/Boxer ³ to translate regulatory statements to semantic representations and then compare the output representations against logical representations in defeasible logic that they created manually. By doing so, they gained a better idea of what each form of representation contains, what is gained or lost, and what next steps are required in order to improve automatic processing of regulatory text.

A framework was introduced in Hassanpour et al., 2011 for the automatic extraction of rules from online text using OWL ontology and Semantic Web Rule Language rules, which is a semantic web rule language combining OWL and RuleML. They identified four necessities to extract rules: (i) ability to recognize domain concepts in text; (ii) recognition relationships between concepts; (iii) assembling of these relationships into chains; and (iv) understanding grammatical structure of sentence to detect relationships. In this thesis, the analysis roughly coincides in the rule extraction phase, but on the other hand, the encoding of the extracted rules with a formal language such as RuleML is not performed. However, it is contemplated in future work.

Araujo et al., 2013 presented a methodology for automatic IE from texts, based on the integration of linguistic rules, multiple ontologies and inference resources, integrated with an abstraction layer for linguistic annotation and data representation. This methodology allows ontology population with instances of

²<https://gate.ac.uk/>

³<https://github.com/valeriobasile/learningbyreading>

events. The methodology has two phases. In the first one, the focus of attention is the corpus study, necessary to build the domain ontology and the linguistic rules. The second phase objective is to integrate linguistic rules with domain ontologies through the use of an inference system and the abstraction layer for linguistic annotation and data representation. The outcome of this phase is a knowledge base composed by the relevant information identified.

Boella et al., 2013 showed a framework to automatically extract semantic knowledge from legislative texts. Instead of using pattern matching methods, they proposed a technique which leverages syntactic dependencies between terms extracted with a syntactic parser. Their idea was that syntactic information are more robust than pattern matching approaches when facing length and complexity of the sentences. They transformed all the surrounding syntax of the semantic information into abstract textual representations from a manually annotated legislative corpus, which are then used to create a classification model by means of a standard SVM system, which is highly related to the rule category classification approach of this thesis.

A three-step acquisition methodology was presented in Lévy and Nazarenko, 2013 to transform the text into a set of self-sufficient rules written in SBVR-SE⁴ controlled language. SBVR-SE stands in an intermediate position between the text and the formal language. The rules were extracted, clarified and simplified at the general regulatory level before being refined according to the business application. In this work and similarly to

⁴<https://www.omg.org/spec/SBVR/About-SBVR/>

Hassanpour et al., 2011, final formal representation takes more importance than in this thesis.

Dragoni et al., 2016 presented a combination of different NLP approaches for rule extraction. The goal of this paper was the same as the ours: an automated rules extraction system that will help in saving time, and it also contributes to a more uniform knowledge representation of formal rules. However, they did not use ML algorithms, which is a key feature of our approach. This work uses the software Boxer logical parser (as in Wyner et al., 2011) and the Stanford syntactic parser⁵; but in this thesis, exploiting statistical semantics is preferred over the logical and linguistic ones.

Finally, Shi and Roman, 2017 proposed another three-step methodology to extract executable rules from NL regulation. Their process was quite manual, while the method presented here tries to automate the whole rule extraction process to achieve a real ACC.

As it can be observed, there is an amalgam of possible solutions for rule extraction. Some of the above frameworks require a domain ontology model which represents the domain knowledge while others not. Some works use ML while others use NLP and grammar-based systems. Although some works try to provide a generic framework or methodology for rule extraction from the regulatory text, there is little, if any, evidence on the practicality of these results when applied to specific real-world case studies in agricultural domain. It is important to remark that although experimental results can be inspirational

⁵<https://nlp.stanford.edu/software/lex-parser.html>

in all these works, the goal of the automated processing of legal texts and the datasets are different; and consequently an experimental comparison with the performances reported in these works is challenging.

It is also essential to note that this thesis is a highly inspired by the Natural Language Understanding (NLU) community, where the key to understand the humans is the ability to analyse the intentions (i.e. rule category classification) of humans and extraction of relevant information from that intention (i.e. rule constraints extraction). Similarly to NLP, NLU uses algorithms to reduce human speech into a structured ontology. Then AI algorithms detect such things as intent, timing, locations and sentiments. However, semantic analysis, the core of NLU, is not yet fully resolved.

Chapter 3

PCT Ontology

The scientist is not a person who gives the right answers, he's one who asks the right questions.

Claude Levi-Strauss

3.1 Introduction

In this chapter, we present the “Pests in Crops and their Treatments” Ontology (PCT-O), which is the basis that supports the rest of the research as it is shown in Figure 3.1. As it was explained in Chapter 1, information related to crop treatments is dispersed and unstructured; therefore, an unified model that supports treatment recommendations is needed; and this is the aim of this chapter.

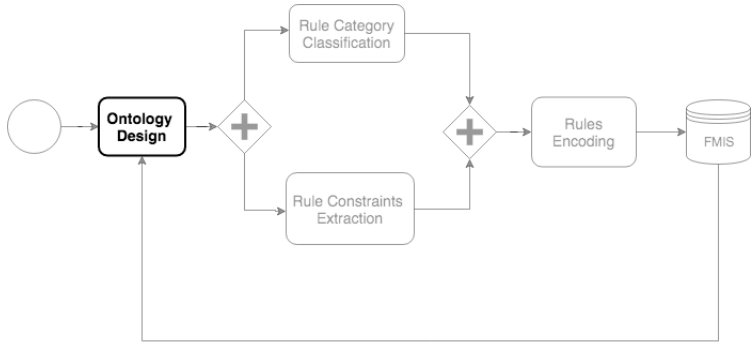


Figure 3.1: PCT-O development: Contextualization of this chapter in the whole workflow.

Although we have focused on Spain data for the rule extraction process, information from other countries could be added. Countries such as U.S., United Kingdom or Canada also provide the information required to populate this ontology in heterogeneous formats, but specific extraction and transformation steps for each new source format would be required. Especially, in the preprocessing phase. On the other hand, the step that align each species/chemical with the selected ontologies and the final integration phase could be reused.

The use of ontologies to model farm related phenomenas is not new in the research. In the biology area, Walls et al., 2012a remark how semantic models facilitate the creation of intelligent applications that manage living species information. The inference capability of ontologies are especially relevant in the

biology area, because it can be used in the taxonomic structures used for classification to simplify conceptual interoperability and data integration. However, the creation of ontologies is challenging due to:

1. Modelling consistence: different interpretations of the selected knowledge area can arise.
2. Data population: data sources are sometimes too complex or too heterogeneous to be processed and may not be added to the model.

Several works in the literature categorise living species, the interactions between them or the effects produced by chemical substances. With respect to living being descriptions, the Integrated Information Taxonomic System (ITIS) (Information and Database, 2010) contains taxonomic information of aquatic and terrestrial flora and fauna, the Catalogue of Life model (Jones et al., 2000) describes 2 million of species, and the NCBI taxonomy (Federhen, 2012) stores the organism names and taxonomic lineages in the INSDC database. All these models provide a comprehensive collection of species but they do not provide very detailed information about their features and behaviour. Other works provide extended taxonomies with further information such as species descriptions, biology, lifecycle, habitat, and interaction with other species. An example of this type of works is Wikispecies (Wikimedia Foundation, 2017), which contains near half a million of species, although the information provided for each species is limited. Another system is the European Nature Information System (EUNIS) (C.E. et al.,

2004), which includes a large collection of species obtained from other databases and indicates the geographical distribution and the level of extinction threat of those species. Another relevant work is the Encyclopedia of Life (Li et al., 2004), which provides more detailed information about a million of species and even a basic description of the interaction between species. However, it does not detail the kind of interaction they have.

Focusing on plants, the U.S. plants database (Natural Resource Conservation Service, 2016) includes a quite detailed textual description of U.S. plants, their distribution, life cycle, and common pests. Sini, 2009 describes the AGROVOC vocabulary, an agriculture thesaurus with a taxonomy of living beings that includes the main used crops and pests in the form of hierarchically related concepts. Finally, the Plant Ontology Consortium, 2002 defines a set of ontologies to describe plants, their genes, diseases and growing process that include the relation between plants and harmful virus and bacteria.

Some works specifically focus on the interactions among species. Rodriguez-Iglesias et al., 2016 propose an ontology that details the pathogens that affect plants. It integrates data related to both plant physiology and plant pathology with the objective of facilitating the interpretation of phenotypic responses and disease processes. Comparable to this, Walls et al., 2012b analyse the infectious diseases of plants and the pathogens that cause them. They reuse vocabularies from other plant, pathogen and disease ontologies such as the Infectious Disease Ontology (IDO) (Cowell and Smith, 2010). Finally, GeoSpecies (DeVries, 2013) relates each concept to the Encyclopaedia of Life, Wikipedia, Wikispecies, NCBI, ITIS, and other similar systems. Instead of

providing proper information about the stored species, it focuses on providing equivalences between the aligned models.

With respect to crop treatments, PubChem model (Fu et al., 2015) describes chemical structures, biological activities and biomedical annotations. This includes pesticides and the environmental effects they produce. However, this information is text-based and it is not linked to any living species model. ChEBI ontology is another model describing chemical substances (Degtyarenko et al., 2008). It contains natural molecular entities and synthetic products that affect living organisms. However, it also lacks a semantic relation with the species affected by each chemical product.

Other works integrate parts of all these and other agricultural aspects together. Damos, 2013 proposes the definition of ontologies that allow describing all the characteristics of cultivations. He also indicates the need to link the created models to other related data collections that complement them. Damos et al., 2017 show an ontology to describe pest and the treatments approved by the Greek Ministry of Rural Development and Food. The core of the ontology contains the pests that are related to the affected crops and existent treatments. On a broader context, Athanasiadis et al., 2009 describe several ontologies for data integration in the agricultural field. Especially relevant is their agricultural activities ontology for crop management. Goumopoulos et al., 2009 describe an ontology for PA. It focuses on describing plants and all the technological and electronic devices that surround them in PA. Finally, Rehman and Shaikh, 2011 describe another PA ontology whose core includes concepts for describing crops and their pests.

3.2 Ontology Design Development

The objective of the ontology proposed in this chapter (PCT-O) is to connect crops, pests and treatments into a unified model. The formal description of living species taxonomies can be managed with the previously described ontologies such as NCBI taxon or GeoSpecies, the description of plant pathologies is covered by Rodriguez-Iglesias et al., 2016 illnesses ontology, and PubChem covers the application of chemical substances. However, they do not model all the crop protection aspects. Specifically, they do not cover the relation between crops, pests that affect them, and the solutions approved by each country to deal with them. Only (Damos et al., 2017) make a proposal to relate information about pests and treatments to the affected crops. However, they propose a high-level model that does not provide detailed properties about each of the proposed classes. The proposed PCT-O allows describing the conditions required by a pest to produce outbreaks and the restrictions on the treatments.

3.2.1 Structure of the PCT-O

The core of the proposed model can be considered as an extension of the disease triangle described in Rodriguez-Iglesias et al., 2016, which consists of a virulent pathogen, a susceptible host and a propitious environment. It has been extended to include non-pathogen pests and the definition of treatments for the pests. We have also modelled the provenance of the information to allow updates and correction of errors in the sources

and in the generation process.

The ontology has been created with the *Methontology* methodology (Gómez-Pérez et al., 2004). Specifically, the modelling has been guided to answer the following questions: Which is the pest that is affecting a given crop? Which treatment do I have to apply to deal with the pest? When do I have to apply the treatment? What are the sanitary/environmental restrictions of the treatment? In the construction process of the PCT-O, there is a special emphasis on reusing existing models to improve the ontology interoperability. Specifically, it is analysed widely used models of living species (which include both crops and pest) and chemical substances.

The core *Species* and *ChemicalSubstance* classes in the model have DBpedia equivalents, and their instances are linked to NCBI taxon, PubChem, ChEBI ontology instances and the Spanish Wikipedia pages (using *owl:sameAs*). The connection between these elements has been guided according to the information provided in the Spanish guides for pest diagnosis and management ¹. The Spanish guides that detail the pest characteristics and treatments have provided us the terminology and relations used to construct the proposed ontology. However, their lack of structure has forced the use of a coarse level of granularity for properties, leaving many of them as simple text fields. Figure 3.2 shows the conceptual view of PCT-O. The main concept is the *Species* concept, which describes the name and characteristics of the included species. It has been specialised into *Crops* grown by farmers and *Pests* that harm the crops. Crops that

¹<http://www.mapa.gob.es/es/agricultura/temas/sanidad-vegetal/>

act as weeds can be classified as both types. The attributes are the common and scientific names, a description, its distribution, images, and equivalency relations with other species models (e.g., NCBI taxon). The *Outbreak* class models the interaction between crops and pests. It contains a textual description of the produced symptoms, the identification and analysis procedures used to establish that a pest is affecting a crop and the existent prevention measures to reduce the risk of infection. The *OutbreakControl* class models the procedure to control a specific kind of *Outbreak* and its location restrictions. Humidity and temperature are the main triggers of outbreaks. Therefore, control procedures and recommendations may vary depending on the climatology of each region. This class includes the period of time in which the pest is harmful to the crop, the description of a way to estimate the infection risk, the description of the best moment to take action to reduce the damages, and the list of treatments approved in the location for dealing with the pest. The *Treatment* class describes four kinds of treatments: *Biological*, *Bio-technological*, *Physical* and *ChemicalTreatment*. Biological treatments make use of predators, physical treatments describe manual measures such as removing infected fruits, biotechnological measures mostly use traps and pheromones, and chemical treatments use pesticides. Each treatment has a description of the treatment itself. The chemical treatments are linked to the pesticides approved by the government (*Pesticide* class), the regulated amount (*doseRange*) and the legal period between the application and the harvest (*securityPeriod*).

The ontology describes the substances dangerous to the environment contained in pesticides through the *ChemicalSub-*

stance class. It includes the common and scientific names of the substances and a description of the effects caused and interactions with other species. We link the substances to PubChem, ChEBI ontology and the Spanish Wikipedia through the *owl:sameAs* property. PubChem link is especially relevant as it contains information about the environmental hazards produced by the chemical substances, and the recommended restrictions of use (e.g. many chemical substances must not be used near water sources or some protected/commercial species). We think this information is vital to be able to select appropriately the least aggressive solution among the existing ones for a given site at a given time.

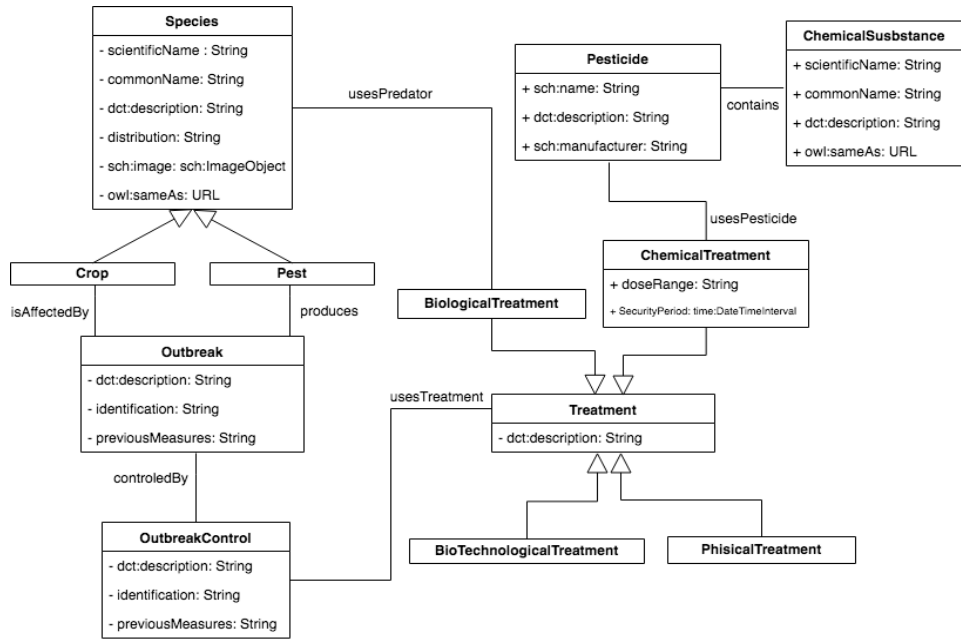


Figure 3.2: Plant affections and their treatment ontology.

3.2.2 **Ontology construction**

The backbone of the ontology instances are the NCBI taxon and the Spanish Wikipedia for living species (crops and pests), and PubChem, ChEBI ontology and the Spanish Wikipedia for pesticide substances. The NCBI taxon, PubChem and ChEBI ontologies are well-known models in their respective fields and provide the scientific names for each element (crop, pest and chemical substances). Specifically, NCBI taxon provides a hierarchy of species appropriate for identification of families of crops. The Spanish Wikipedia provides alternative scientific and common names that are helpful in the disambiguation process. Each model has additional information about species and chemical substances such as taxonomic relations, definitions, chemical formula and so on. We do not currently use this information, but the linkage makes it accessible for future improvements. To populate the PCT-O we have focused on the official information about crops and authorised pesticides maintained by the government of Spain. This section describes the data sources, the ontology construction and the process developed to extract the available information and represent it according to the ontology model.

3.2.2.1 **Tools used for ontology construction**

We have selected OWL (McGuinness and Van Harmelen, 2004) as the description model for our ontology and its instances. OWL is the most common RDF-based description model in the semantic field and it enriches the description capabilities

of RDF/RDFS (Brickley et al., 2014) by supporting complex relations between classes and detailed characterisation of properties. The construction of the ontology has required the use of multiple tools and libraries to define the model and populate it from the selected sources. The ontology has been created using the Protégé editor ², a tool designed to facilitate the creation of OWL schemas. With respect to the ontology population, it has required the extraction of information from multiple PDF files. This has been done using Apache PDFBox ³, a Java library for PDF processing. For the processing of the extracted content, a workflow that fills an Apache Jena ⁴ triple-store (a RDF database that support named graphs) has been created using Spring Batch ⁵.

3.2.2.2 Data sources used for information extraction

The description of the effects that each pest has in each crop and the processes established to detect and treat them have been obtained from the following heterogeneous document collections provided by the Spanish Ministry of Agriculture:

1. The laboratory diagnosis sheets of noxious species for crops created by the phytosanitary diagnosis and survey laboratory, which is a collection of 464 scanned PDF documents describing plants, insects, bacteria and virus (scientific and common names of the pests that affect crops, their

²<https://protege.stanford.edu/>

³<https://pdfbox.apache.org/>

⁴<https://jena.apache.org/>

⁵<https://spring.io/projects/spring-batch>

distribution in Spain, symptoms, detection measures and identification procedures);

2. The guides for the integrated control of pests created by the national plan for sustainable use of pesticides, which is a collection of 21 digital PDF documents that describe the crops affections in Spain and the recommendations for their treatment (common name of the crops, the common and scientific name of the noxious species, control and prevention measures, and available non chemical treatments); and
3. The registry of pesticides approved by the national institute for agrarian research and technology, which is a repository containing 2,426 records detailing the pesticides allowed in Spain, their composition and use restrictions. This source will be explained deeply in Chapter 4.

The content of these sources connects the living species information with the chemical substances used on them. The main issue of these collections is their heterogeneity. None of these data sources is completely structured and uniform. Some parts have a tabular structure, but most of them are described as paragraphs of plain text. The text sections are similar between documents but not exactly equivalent. Additionally, the quality of several scanned documents is low, making data extraction burdensome.

3.2.2.3 Preliminary information extraction

As it has been explained in previous sections, the information extracted in this chapter is coarse-grained and it will be refined in following chapters to achieve ACC. We have followed the population process described in Figure 3.3. The first step has been to extract the textual content and accessible images from the source PDF files. Then, an IE step for each source has been applied to identify the elements required in the ontology. Textual content is used for filling the different properties of the instances, while the images are stored as a graphical representation of each concept. All the extracted images are stored, independently of the relevance of their content. To simplify data integration, each extracted resource is aligned to the previously described ontologies using the common and scientific name of crops, pests and chemical substances as matching text. Having identified the species/chemical substances in the resources, their integration is direct. The first half of the process is dependent of the selected sources, but the second half can be directly used for integrating future supplementary data collections. In the data extraction step, if the origin of the PDF file is analogical (scanning of a printed document), the OCR process in the PDFBox library is applied to extract the text. However, scan quality of the source files limits the quality of the extracted content. Most of the extracted text contains minor errors due to bad recognition of some characters, but a few have higher error rates. In addition to this, the non-plain text parts of the documents are not correctly extracted due to PDFBox limitations (e.g., captions of photos or tabular information).

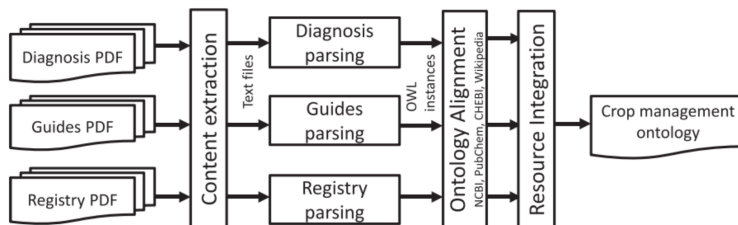


Figure 3.3: Ontology population process.

The parsing step makes use of the fact that all the analysed sources are divided into sections whose content mainly corresponds with properties of the defined model. It identifies these sections according to a list of predefined headers for each type of document that contain all the variant forms found for the sections names and structure of the source documents. Additionally, we have defined specific rules containing syntactic patterns describing textual constructions in the documents when describing the common or scientific name of a species. The extracted information and its provenance information is stored according to the PCT-O model. The alignment step matches the extracted resources describing species (crops and pests) with the NCBI taxon and the Spanish Wikipedia, and the chemical substances with respect to the Spanish Wikipedia, PubChem, and ChEBI ontologies. The alignment of the species is used to directly merge the information of the involved data collections. The alignment of the chemical substances is used to facilitate the identification of equivalences between the different products

used to deal with the pests.

The alignment has been performed looking for equivalences in the scientific names of species and chemical substances contained in the documents. The complexity of this alignment process has come from the need of identifying and correcting the errors in the sources, and because of the existence of synonyms and variants of names of the living beings and chemical substances. To deal with these problems, we have performed the following alignment sub-steps.

1. First, we have extended the applicable synonyms and variant names for each extracted crop/pest with further names obtained from the Spanish Wikipedia. This has been done looking for the common names in the Spanish Wikipedia and extracting the scientific ones contained in the corresponding info-boxes.
2. Then, all the scientific names are matched (exact match) with the corresponding ontology/model (NCBI, PubChem, ChEBI). If a match is found, the alignment is established. If there is no correspondence, we have used the Levenshtein distance (Levenshtein, 1966) to identify matches with minor errors and variants of the scientific names. For this comparison, the scientific names are normalised removing abbreviations, numbers, and texts in brackets. Name heterogeneity has led us to use a threshold of 20% of the name size to decide if the most similar name can be aligned or not. Therefore, shorter names allow smaller differences than longer ones. This threshold has been selected experimentally to reduce the number of incorrectly

aligned concepts (we prefer to leave them unaligned).

3.3 Results

The resulting ontology consists of 549 pests that affect 462 crops through 3,471 outbreaks. Figure 3.4 shows the pests in the model aggregated by family. It can be observed that most of them are fungi and arthropods. In addition to those, there are virus, bacteria, nematodes and other plants. A few pests are from species that do not fit in the previous categories. To deal with these pests, there are 42,397 different chemical treatments involving 2,109 pesticides with 566 different chemical substances, and 219 alternative treatments.

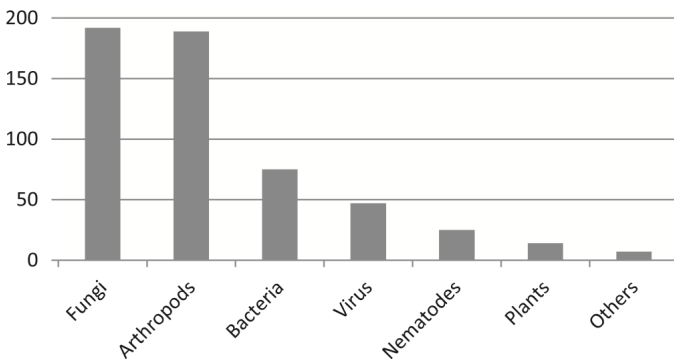


Figure 3.4: Classification of pests.

A manual review of the ontology has shown that 96.12% of the species (pests and crops) have been correctly aligned to their scientific name in NCBI Ontology. The main source of errors are problems in the description of the names of the sources (e.g., “summer cereals”), the use in the sources of the fruit name instead of the plant name or the lack of equivalences for some of the used common names. We have also reviewed the quality of the extracted description of the species, the symptoms and the information related to prevention and intervention time. Here the quality is worse due to the difficulty of extracting the content. There are almost no records without syntactic errors. Most of them are small, but to be usable, it is required to correct them through a manual proofreading. Something similar happens with treatments: the extracted information has been correctly assigned to the corresponding concepts in the ontology, but there are many syntactic errors caused by the extraction. Finally, we have also reviewed the alignment of the chemical substances with the ChEBI database (PubChem is linked to it). The result shows that just 59.9% of the chemical substances have been correctly aligned, 27.7% of them are left unaligned and the rest (12.4%) are incorrectly aligned. This alignment problem is caused by the lack of correspondence between the Spanish common/scientific names for the chemical substances in the sources and the Spanish Wikipedia. The Spanish Wikipedia has proven to be a good source to align common and scientific names of living species but its coverage for chemical substances is much worse. It does not describe many specific substances, thus the Spanish names cannot be aligned to the English ones in the selected ontologies.

From these data, it can be observed that current crop protection is completely focused around the use of chemical products. There are many more chemical solutions than alternative ones, and their amplitude of action is also broader because they affect numerous pests. With respect to alternative approaches, they are only able to deal with a small set of the pests (mainly insects) but they do not have secondary effects for humans or nature.

3.4 Discussion

As indicated in the state of the art section, there are a few models for the description of species and chemical substances, but only Damos, 2013 and (Damos et al., 2017) provide some relation between crops, pests, and treatments. PCT-O goes a step further by including the description of the conditions of these relations. Therefore, in PCT-O, it is possible to specify the period of time when a pest is harmful, when it is needed to react, and the nature of the treatments. The next closest solution is the PubChem ontology that describes thousands of chemical substances and for the suitable ones, it indicates the common name of the crops to which the substance can be applied according to USA legislation. However, it is not linked to any species ontology and may be ambiguous. Additionally, it indicates neither a detailed list of the noxious species the chemical substance can deal with, nor the symptoms, periods of control or chemical alternatives. In the analysed scenario, we have shown how PCT-O helps in terms of interoperability and data integra-

tion between crops, pests and treatments information. Thanks to it, it is possible to construct a semantic recommendation system that helps to determine the pests that affect each crop and how to treat them according to the official Spanish regulations and guidelines. Moreover, ambiguity among different countries is removed because the crops, pests, and pesticides are linked to commonly used ontologies and taxonomies.

On the other hand, according to the cyclic nature of the MATTER methodology followed during this thesis, we need to extend the PCT-O in order to represent the complexity of a chemical treatment application. According to some related works (Dragoni et al., 2016), modelling will consist on categorising the rules that prescript the chemical treatment application and the extraction of the different constraints that must be taken into account by the FMIS in order to prescribe the correct pesticide.

With respect to create models from open data sources, there are some issues that are relevant to remark because they illustrate the complexity of obtaining a complete model from the available official sources. On the one hand, data quality has been an issue that has complicated the IE and it has added errors. We have found considerable cases where a correct equivalence has not been found and chemical substances have been incorrectly aligned. The cause of this is mainly due to the incompleteness of Spanish Wikipedia in biology/chemistry area and the similarity between some scientific names of species/chemical substances.

Regarding data sources, it also arises another issue that points to the completeness and overlap of the data. Each data

source was created by its producer with a different purpose and they do not completely overlap. For instance, the guides only cover a subset of species described in the diagnosis files. As a result, the populated ontology does not have a uniform coverage: some species are very detailed, other ones contain very limited information. These restrictions reduce the usability of the extracted information, but it is a good starting point that will be revisited in Chapter 5.

All these issues lead to a limitation of PCT-O is the semantic granularity of the model and its suitability to solve ACC. The information contained in fields such as pest description, control period, identification procedures, or intervention time is described as plain text, so queries on these fields are imprecise and human intervention is necessary. For example, when querying for “Brown leaves” as pest symptom, pests that only produce brown leaves in some specific situations will be returned with the same importance than pests with brown leaves as representative symptom. Solving this problem would require again to extend the ontology to allow a precise description of such content.

Regarding to the ACC aim, some descriptions are quite clear (e.g., temperature under 25 degrees), but others, as indicated in Nash et al., 2011, need human interpretation (e.g., high temperature). In this situation, a semantic baseline for each crop must be defined to allow the mapping of all the imprecise descriptions to measurable values that could be compared to sensor values within a PA context. We have done a preliminary processing to identify the common temperature and humidity patterns in the source documents and more than 80 different rules have been

needed. Additionally, we have to perform approximations that are crop and pest dependent. For instance, many documents say that a crop is vulnerable to a pest with high temperature, but how much temperature is “high”? To model it semantically, this must be translated to a numerical range (as it is in many other descriptions). However, with the source information alone it is not possible to determine a precise value, and an approximation must be given. Another source of ambiguity is the period of control of a crop. The growth stage is sometimes properly described (e.g., flowering), but other times it is referenced using periods of months or seasons (e.g., May). This must be interpreted depending on the place and the climate conditions of a given year.

3.5 Summary

This chapter proposes the PCT-O ontology, a model to describe the outbreaks that pests produce to crops and the approved ways to treat them. Currently, there are a few ontologies to describe taxonomies of living beings but none allows describing their interrelations as the PCT-O ontology. The information extracted based on this ontology, could be used as a recommendation system that helps to identify the pests affecting a crop and their legal treatments. The information has been extracted from official information in Spain about crops, pests and approved treatments. This process has been complex due to the heterogeneity, format and quality of the data sources. The extraction and source errors, complemented with synonymy and

name variants, have forced us to use a disambiguation process of scientific names based on the alignment of species and chemical substance records with ontologies such as NCBI, PubChem, ChEBI and Wikipedia. The model can be used for tasks such as the identification of outbreaks, identification of site-specific related conflicts with the treatments, and comparison of solutions between country legislations.

According to the MATTER methodology followed during this thesis, a refinement of the ontology and the IE will be undertaken. For example, after reviewing the guidelines, we have observed that current crop protection is completely focused around the use of chemical products. Therefore, an extension of the *ChemicalTreatment* class alongside an improvement in the extraction of the rules that orchestrate the treatment process. It will be done through a two-step process: rule category classification (Chapter 4) and rule constraints extraction (Chapter 5).

Chapter 4

Rule Category Classification

Physics is like sex: sure, it may give some practical results, but that's not why we do it.

Richard P. Feynman

4.1 Introduction

Following the framework described in Nash et al., 2011, in this chapter it is presented a rule category classifier development by evaluating the combination of NLP methods, resampling techniques and ML algorithms. The final result is a program that can automatically discern between prohibitions and obligations in the agricultural domain. We have developed a gold corpus to train and evaluate the rule classifier because, to the best of our knowledge, there is no available gold corpus focused on phytosanitary regulations. With this chapter, we provide in-

sights about the possibilities and limitations of existing ML, resampling and NLP techniques for its usage in agriculture for supporting the development of decision support systems and FMIS. We also present some preliminary results comparing traditional ML algorithms and DL algorithms. It is important to remark that in our proposed rule extraction system, the correct operation of the classification component is crucial because the meaning of the rule is inverted. For example, a pesticide that is forbidden could be prescribed as permitted. In Figure 4.1, it is contextualised the rule classification inside the whole development of a rule extraction system we are proposing in this thesis.

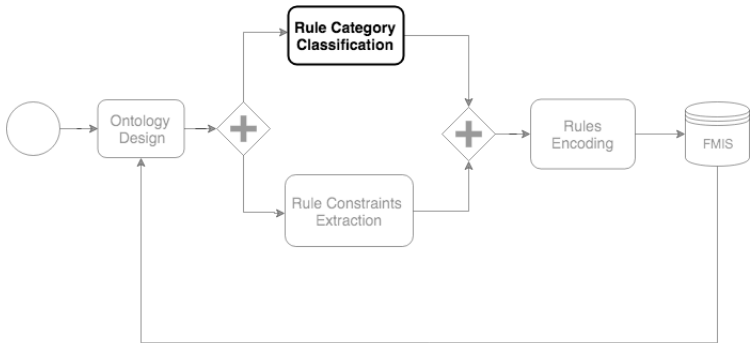


Figure 4.1: Rule Category Classification: Contextualization of this chapter in the whole rule extraction workflow.

Text or sentence classification is one of the principal tasks

of ML and refers to the process of designing proper algorithms to enable computers to extract features from texts and classify them automatically. Sentence classification can be seen as a grammar-acquisition task, where the performance can be measured as the percentage of correctly parsed sentences and the percentage of correctly rejected non-sentences. In the first days of the domain, people research text classification through manual classification rules according to linguistic rules and grammars; but currently, research is centered in computer-based automatic classification with ML techniques. Before the emergence of DL, the traditional pipeline to automatically classify text has two main steps: (i) preprocessing to extract features through different NL modules and (ii) training with ML algorithms. Examples of this pipeline, shown in Figure 4.2, are present in many works.

For example, in Zhang, 2004, they used the SVM algorithm for multi-class text classification. The method mainly leverages the vector space model as a feature, which transforms the document into a high dimension sparse vector per the features of the text and then enters it into the SVM classifier. However, the results show that in the context of large data volumes, classification accuracy of automatic approaches is much better than the expert definition of the rules.

In Khoo et al., 2006, they presented a set of experiments involving sentence classification, addressing issues of representation and feature selection, and comparing their findings with similar results from work on the more general text classification task. Their investigations compare the use of various popular classification algorithms such as SVM and NB with various popular feature selection methods. Their results showed the su-

periority of SVM, and a not very relevance (even detrimental effect) of some feature engineering techniques such as stop-word removal and lemmatization.

In Palkar et al., 2016, the researchers provide a comparative study of multiple well-known supervised ML algorithms on three standard datasets confined to the domain of movie reviews. In the study, they classify sentiments by training and evaluating SVM, RF, Logistic Regression and naive Bayes algorithms. This is approach is quite akin to the ours. In their results, SVM and RF seem to have the best performance. They also consider to use DL in the future.

In Siddiqui, 2016, it is shown a comparative study for text classification using SVM, KNN and naive Bayes. They used different datasets related to the task of sentiment analysis. After executing their experiments, they found that depending on the dataset and preprocessing techniques, different approaches could be more or less successful. On the other hand, in a overall setting, SVM could be considered the best approach.

Currently, text classification has a wide range of domains where it can be applied. For example, in the field of journalism, publications need to be classified according to the columns; in mail processing tasks, the mail system classifies contents of the messages to determine whether the message is spam or not. Moreover, currently it is quite related to problems such as intent classification in personal assistants.

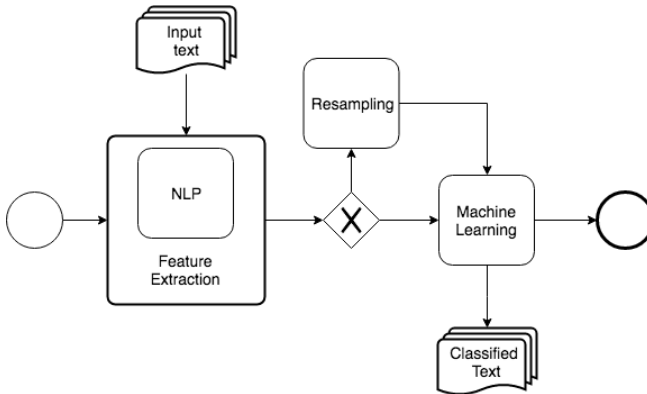


Figure 4.2: Traditional rule classification pipeline.

4.2 Rule Classifier Development

In this section, we explain how the rule category classifier is developed. Three steps are covered. Firstly, the preprocessing phase where the words are transformed into numeric values; secondly, the data augmentation step, where the dataset is rebalanced due to the imbalanced nature of prohibitions and obligations in our corpus; and thirdly, the ML algorithms training in order to obtain a suitable rule category classifier.

4.2.1 Preprocessing

Since the dataset is formed by words, it is necessary to convert them into numeric values; and moreover, it is also necessary to rebalance the dataset to avoid statistical bias due to the high amount of obligations compared to the number of prohibitions.

4.2.1.1 Feature Engineering

Rule-based and weighting A preprocessing step using NLP techniques is necessary to extract the most important words or groups of words from inside the rules and improve the performance of the classifier. As Collobert et al., 2011 explained, the choice of the optimal text preprocessing technique is an empirical process that is mainly based on linguistic intuition followed by trial and error. In our experiments, we use the following NLP techniques to improve the ML process by adding linguistic knowledge:

1. Part-of-speech (POS) tagging
2. Stemming
3. N-grams
4. Feature Weighting

Since it is widely used, we use the Stanford POS tagger in this thesis (Toutanova et al., 2003). For stemming, we use the Porter algorithm (Porter, 1980). Within the n-grams approach, we use both unigrams and bigrams. Finally, to provide a weight for

each word or group of words in the corpus we use the term frequency-inverse document frequency (*tf-idf*) (Raschka, 2014) because it decreases the weights of words that are not relevant and not present in the list of stop words.

Word Embeddings As explained in the previous chapters, word embeddings replace the hard matches of words in the NLP based approach with the soft matches of continuous word vectors while the multiple layers of hidden vectors created by deep neural networks further abstracts the embeddings to obtain the underlying feature representations. Consequently, word embeddings mitigate the unseen words problem of the feature-based models while the whole DL models help to avoid feature engineering and provide effective feature representations. Although this approach should overcome the traditional rule-based NLP, some factors such as the lexical variability of the dataset can harm the performance.

Since word vectors are very computationally intensive to train on, it is often convenient to use word vectors which have been pre-trained rather than training them from scratch for each project. These models enables anyone building a ML model involving NLP to use this readily-available component saving time and resources that would have gone to training a language-processing model from scratch. Under this context, linguistic units are initialised with embeddings that are pre-trained from extremely huge amount of unlabelled text. In this thesis, we have used a dataset contains 1,000,653 word embeddings of dimension 300 trained on the Spanish Billion Words Corpus. In

this work, we use pre-trained Word2vec with Skip-gram (Rong, 2016) to initialise word embeddings. The whole statistics are shown in 4.1 and 4.2.

In summary, all the evaluated models are based on two text representations: Word2vec word embeddings and *tf-idf* representation.

Table 4.1: Skip-Gram Algorithm Parameters.

Skip-Gram Algorithm Parameters	
Algorithm	skip-gram model with negative-sampling
Minimum word frequency	5
No. of noise words (Negative Sampling)	20
No. of noise common downsampled words	273
Embedding Dimension	300

Table 4.2: Original Embedding Corpus Statistics.

Original Corpus Statistics	
No. of Raw Words	1,420,665,810
No. of Sentences	46,925,295
No. of unique tokens	3,817,833
After Skip-Gram Corpus Statistics	
No. of Raw Words	771,508,817
No. of unique tokens	1,000,653

4.2.1.2 Data augmentation

In contrast to rule based systems, further challenges come from the usage of ML techniques due to its statistical nature. It has been reported that one of these aspects is related to class imbalance, in which examples in training data associated with one class heavily outnumber the examples from other classes (Japkowicz and Stephen, 2002; Chawla et al., 2004). In our corpus, this problem arises because we have many more obligations than prohibitions. In this situation, the ML system may have difficulties learning the concepts related to the minority class (prohibition in our case). Despite its shortcomings, one of the procedures that has been applied in many studies is resampling (He and Garcia, 2010). Resampling is performed by oversampling or undersampling data to change the frequency of classes in the training data extracted from the gold corpus in proportion to a cost model. Resampling is only applied to the training set because the test set must be kept in its original state. In this part of the work, we perform a broad experimental evaluation involving five different resampling methods:

1. Random oversampling (ROS),
2. Random undersampling (RUS),
3. SMOTE,
4. ADASYN,
5. Tomek links

In ROS, the minority class is randomly replicated to force the learning algorithm to correctly classify instances of that class, whereas RUS involves the random deletion of examples of the most frequent class to yield the opposite result. SMOTE is an advanced method of oversampling developed by Chawla et al., 2002. This approach aims to enrich the minority class boundaries by creating artificial examples in the minority class rather than replicating existing examples to avoid the problem of overfitting. ADASYN, presented by Skalidis, 2016, is another method of oversampling. The essential concept is to use a weighted distribution for different minority class examples according to their level of difficulty in learning, where more synthetic data are generated for minority class examples that are harder to learn compared to those minority examples that are easier to learn. Tomek links is a method of undersampling that searches for instances of closest neighbours that do not share the same class label (Tomek, 1976). When this relationship is identified, the Tomek link is removed from the data set. This process is repeated until no more Tomek links can be found.

Additionally, we provide some insights with some preliminary experiments using a resampling technique consisting in replace words in the prohibition class based on the nearest neighbour algorithm and the word embeddings presented in the previous section.

4.2.2 Machine Learning Training

After preprocessing using the feature engineering and resampling techniques, we must apply ML algorithms to obtain a rule classi-

fier that can discriminate between prohibitions and obligations. The term ML refers to the automated detection of meaningful patterns in annotated data. Figure 4.2 shows the traditional pipeline, where feature extraction through traditional NLP techniques are applied.

4.2.2.1 Shallow Machine Learning Models

The specific ML algorithms used to train the rule category classifier are:

1. Logistic regression,
2. SVM,
3. Naive Bayes, and
4. Random Forest

The first three methods are chosen because they generate linear models that generally yield good results in high dimensional sparse problems, such as text classification, that overcome the issue of dimensionality (Bellman, 1961). A RF method is chosen due to its effectiveness when applied to different problems, and contrary to linear classifiers, it can learn complex models that are sometimes necessary to correctly describe a classification problem. If the performance of linear and non-linear classifiers is the same, linear classifiers are typically selected because they are simpler than nonlinear classifiers.

4.2.2.2 Deep Machine Learning Models

When a DL architecture is used, the process for classifying rules is slightly different as shown in Figure 4.3. The main distinction is that word embeddings take importance in this approach in order to maximise the DL algorithms performance. The dominant approach for many current NLP tasks are RNNs, in particular LSTMs.

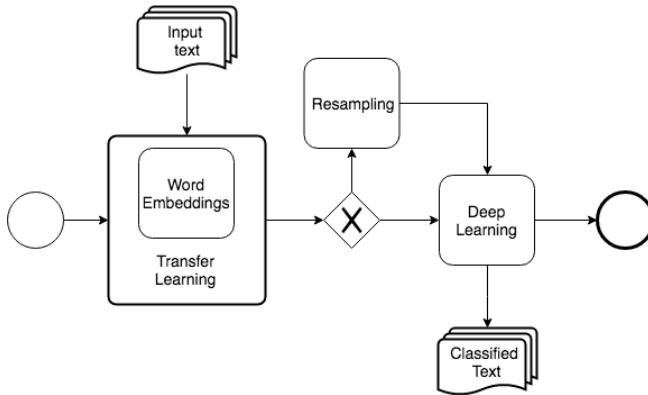


Figure 4.3: Deep rule classification pipeline.

In this preliminary experiments, a single deep neural network architecture where three architecture decisions are evaluated: word representations, representation aggregation and word sequence representations. Regarding the word representation, we evaluate the use of transfer learning (initialised word embed-

dings) and not using it. About, the aggregation component, we evaluate using the output state of the LSTM and aggregating the internal states through a concatenation of an average pooling and maximum pooling of them (Max-Mean). While in the case of word context modelling, it is evaluated the use of simple LSTM in comparison to BLSTM. Thereafter, on top of this layer, we use a sigmoid inference layer. This is an experimental constant. The output of the architecture is the rule category: prohibition or obligation.

4.3 Experiments

In this section, we collect the necessary data to train the ML algorithms; and evaluate their suitability for ACC.


4.3.1 Local structure of agricultural standards in Spain

As explained in a preliminary way in the previous chapter, in Spain, the documents containing the phytosanitary products that are allowed and how to apply them to comply with environmental regulations, are published in the Spanish official registry of phytosanitary products and currently contains 2,426 documents in PDF format. In Figure 4.4, it is shown an example of one of these documents. The part boxed in dark blue shows that this document is official and published by the Government of Spain. The light blue inset contains the table that includes the information of how the pesticide must be applied. Inside

this table, we can distinguish between two parts: the first one (green) shows the chunk of the regulation that is structured and that could be easily transformed into a machine-readable format using different heuristics. Here, we can find the crop, the plague and the dose that must be applied. On the other hand, the element boxed in red colour is formatted with unstructured NL and its translation into a formal rule is the motivation of this research. In this part, we can find different spatiotemporal constraints that currently cannot be easily extracted. Each of these constraints can be categorised as an obligation or a prohibition.

Below, there are some examples of rules (translated into English) that appear in these documents with their categorisation (obligation/prohibition). These rules will be used to train and evaluate the ML techniques that are the base of the final rule classifier.

- “Apply only until flowering” (Obligation)
- “Treat from the stalk develops until the ear emergence” (Obligation)
- “Do not apply in crops with fruits destined to preserve” (Prohibition)
- “Never apply after 10 leaves” (Prohibition)


Nº REGISTRO: 11826
APOX

Usos y dosis autorizados:

USO	AGENTE	Dosis %	FORMA Y ÉPOCA DE APLICACIÓN (Condic. Específico)
Alcachofa	PULGONES	0,1	Realizar un única aplicación con un volumen de caldo máximo de 300 l/ha, a partir del estado vegetativo de 9 o más hojas desvolgadas.

Figure 4.4: Example of document from in the Spanish official registry of phytosanitary products.

4.3.2 Gold Corpus Creation

A gold corpus is a set of annotated texts that serves as a basis for the training and evaluation of ML algorithms. Currently, to the best of our knowledge, there is no available gold corpus focused on phytosanitary regulations and, therefore, we have developed our own corpus by using the MATTER methodology. The corpus is a monolingual Spanish corpus made of 2,426 PDFs harvested from the Spanish official registry of phytosanitary products. We have manually annotated 1,135 rules in NL as obligations or prohibition when the text conveys such meaning related to a phytosanitary product application. Some examples are shown in the previous section. Corpus statistics are shown in 4.3. We think that the corpus has adequate size for the evaluation of algorithms due to the small number of distinctive rules and the

standardised nature of the phytosanitary vocabulary. This statistics clearly demonstrates that the data is imbalanced and the necessity to be balanced in order to get the best performance.

Table 4.3: Corpus Statistics.

Corpus Statistics	
No. of rules	1,135
No. of obligations	1,119
No. of prohibitions	16
Rule length average	22 words
No. of words	25,420
No. of unique words	2,689

4.3.3 Preprocessing

Documents from the Spanish official registry of phytosanitary products contain noise (Figure 4.4), e.g, section headings with repeating punctuations and abnormal text formatting, unexpected line breaks, tables wrongly encoded; and existing standard tokenisers such as the ones that NLTK provides, fail to produce accurate results. For this reason, we instead have built an ad-hoc rule-based tokeniser that processes the documents.

4.3.4 Rule Classifier Evaluation

The evaluation technique measures the correspondence between the results that the classifier generates and those of the gold standard. There is no single evaluation metric that is right for

any classification problem. In practice, it is always recommended to compare different classification models on the particular dataset taking into account different metrics. Moreover, it is valuable to think the high-level goal of the application: The FMIS where the rule classifier could be integrated must classify accurately the maximum number of rules to reduce the risk of prescribing the wrong pesticide or application. This goal can be evaluated with three metrics: recall, precision and a combination of both of them: F_1 score.

Recall is a widely used ML metric. In our work, it is defined as the fraction of “truly” prohibition rules that are effectively classified as prohibitions ($n_{pr \rightarrow pr}$) and thus provides a measure of the “completeness” of the system (Eq. 4.1). Recall decreases if the number of prohibitions misclassified as obligations ($n_{pr \rightarrow ob}$) increases. If recall is 100%, it means that no prohibitions has been classified as obligations.

$$Recall = \frac{n_{pr \rightarrow pr}}{n_{pr \rightarrow pr} + n_{pr \rightarrow ob}} \quad (4.1)$$

Precision is another widely used metric and, loosely speaking, provides a measure of the “soundness” of the system. More specifically, it is the proportion of the well-classified rules as prohibitions ($n_{pr \rightarrow pr}$) to the total number of rules classified as prohibitions ($n_{pr \rightarrow pr} + n_{ob \rightarrow pr}$) as shown in Eq. 4.2. Precision decreases if the number of obligations misclassified as prohibitions ($n_{ob \rightarrow pr}$) increases. In this work, if the precision is lower than 100%, it means that some obligations are classified as prohibitions and a rule such as “Apply this pesticide in Spring” could be interpreted as “Do not apply this pesticide in Spring”.

$$Precision = \frac{n_{pr \rightarrow pr}}{n_{ob \rightarrow ob} + n_{ob \rightarrow pr}} \quad (4.2)$$

Greater recall and precision values indicate better performance, however, it is important to highlight that there is a trade-off between optimising recall and optimising precision. So while precision and recall are very relevant metrics, looking at only one of them will not provide the full picture. Finally, the F_1 score combines precision and recall trying to provide a single metric to ease algorithms comparison as shown in Eq. 3. In this work, this is the measure that will serve to decide which is the most balanced approach and probably the best approach to categorise rules.

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \quad (4.3)$$

Although, accuracy is a well-known standard measure for measuring classification performance it is not recommended for datasets with skewed distribution this measure can be misleading.

4.4 Results

In this section, we divide the results obtained with the traditional shallow ML models such as SVM and logistic regression; and those obtained from the deep models.

4.4.1 Shallow Machine Learning Results

This section shows experimental results of 96 different combinations of ML algorithms, resampling and NLP techniques used to build the rule classifier. All of them are the averages of 30 runs. In each of the runs, we use stratified 10-fold cross validation to find the best hyperparameter settings used in ML algorithms (Table 4.4). This statistical technique provides good performance estimates with minimal assumptions and making results less prone to random variation. The main disadvantage of cross-validation is increased computational cost, but in this phase of the research, it is more valuable to obtain a good performance estimation.

The implementation of the machine learning algorithms is the one provided by Scikit-learn¹ (Pedregosa et al., 2011), one of the best known and most widely used machine learning libraries. This package, written in Python, includes the implementation of many popular machine learning algorithms, and preprocessing and evaluation capabilities. The version of Scikit-learn used in this work is 0.19.1.

¹<https://scikit-learn.org/>

Table 4.4: Parameter specification for the algorithms.

Algorithm	Parameters
Naive Bayes	-
SVM	Kernel = Linear C=10 Tolerance = 0.001 Shrinking = true
Random Forests	Estimators = 20 Pruned = false Impurity = Gini
Logistic Regression	Penalty = l2 C = 10 Tolerance = 0.0001

We investigate the learning algorithms in combination with different NLP and resampling techniques in order to find the combination that allows the most accurate rule classification between prohibitions and obligations. There are more algorithms and NLP techniques that are out of the scope of this work, but in future experiments they should be studied to find better approaches.

In table 4.5 we can observe the top 10 combinations of NLP, resampling and machine learning techniques that present the best precision recognising prohibitions. This means that these combinations are the ones that minimise the false positive error ($n_{ob \rightarrow pr}$). In other words, the number of obligations that are

Table 4.5: Summary of the algorithms with best prohibition precision.

NLP	Resampling	Algorithm	% Precision	% Recall
POS	None	Logistic	85	58.57
POS	TomekLinks	Logistic	84.46	60
POS	ROS	RF	84.04	47.85
POS	TomekLinks	RF	81.54	40
POS	None	RF	78.72	34.28
POS	SMOTE	RF	75.73	50.71
POS	ADASYN	RF	74.25	47.14
Bigrams	SMOTE	SVM	72.12	61.42
POS	None	SVM	67.72	52.14
Bigrams	ROS	SVM	67.15	70

classified as prohibitions is minimum. On the other hand, they achieve a low recall, which means that some prohibitions are “lost” and they are classified as obligations ($n_{pr \rightarrow ob}$). It can be seen that POS tagging is the best technique if we want a high precision. Otherwise, we find more diversity in resampling techniques and machine learning algorithms. Maybe, logistic could be seen as the best approach, taking into account that top two results use this algorithm.

Table 4.6 shows the top 10 combinations of NLP, resampling and machine learning techniques that show the best recall recognising prohibitions. These combinations are the ones that

minimise the false negative error ($n_{pr \rightarrow ob}$). In other words, the number of prohibitions that are classified as obligations is minimum or zero in the case of 100% recall. On the other hand, they achieve a very low precision, which means that many obligations are “lost” and they are classified as prohibitions ($n_{pr \rightarrow ob}$). It can be seen that stemming and unigrams are the unique NLP technique that seem to achieve top performance in recall. It is important to highlight, that best results are always achieved by resampling techniques, more specifically oversampling techniques. This is expected because resampling techniques are used precisely to improve the capability of the machine learning algorithms to recognise prohibitions. The problem of these approaches is that because there are so many obligations, if an algorithm is biased to classify rules as prohibition, precision can decrease a lot (the best precision is 23.09%).

Finally, table 4.7 shows the top 10 combination of NLP, resampling and machine learning techniques that shows the best F_1 score recognising prohibitions. These results show the most balanced approaches. These means that if we have no preference about type of error and misclassifying obligations and prohibitions is equally important, these approaches should be chosen. The most balanced combination appears in the first row and it contains POS tagging, Tomek Links and Logistic. This approach obtains a 68.08% in F_1 score.

Observing the rest of the results, it can be seen that POS tagging is in top 3, and Logistic appears in the best two results.

In order to confirm that the approach with best performance is not due to chance, we have computed statistical significance test using the second best approach (POS tagging, ROS and

Table 4.6: Summary of the algorithms with best prohibition recall.

NLP	Resampling	Algorithm	% Recall	% Precision
Unigrams	ROS	Logistic	100	23.09
Stemming	ROS	Logistic	100	21.04
Unigrams	ADASYN	Logistic	100	20.71
Unigrams	SMOTE	Logistic	100	20.58
Stemming	ADASYN	Logistic	100	19.83
Stemming	SMOTE	Logistic	100	19.67
Unigrams	RUS	Logistic	100	6.85
Stemming	RUS	Logistic	100	6.55
Unigrams	RUS	Bayes	100	5.63
Stemming	ROS	Bayes	100	5.54

Logistic regression). The test has been performed using the Welch's t-test (Welch, 1951). We have set the confidence level to 0.01. According to the test, it exists statistical significance between the two approaches, and therefore, we can confirm that the correct selection of NLP, resampling and machine learning algorithm is important to build the most accurate rule classifier.

It is also important to note that in the three tables, Logistic Regression has been the best machine learning algorithm. The rationale of these results is that simple linear models can obtain good results in combination with different resampling and NLP techniques. In order to have more clues about which techniques work better for the rule classification, we are going to visualise

Table 4.7: Summary of the algorithms with best prohibition F_1 score.

NLP	Resampling	Algorithm	% F_1
POS	Tomek Links	Logistic	68.08
POS	ROS	Logistic	67.72
POS	None	RF	67.04
Bigrams	ROS	RF	66.64
Unigrams	ROS	RF	66.54
POS	SMOTE	RF	65.91
Bigrams	SMOTE	RF	63.41
Stemming	ROS	SVM	60.53
POS	ROS	SVM	58.55
POS	SMOTE	SVM	57.39

the results after aggregating all the F_1 score values for every NLP, resampling and machine learning techniques.

Figure 4.5 shows a comparison of NLP techniques without taking into account the rest of the classification components. We can observe that POS tagging is the technique that achieve the best performance. The rest of the NLP techniques obtain similar results, and therefore we can infer that stemming and bigrams have little influence in F_1 score.

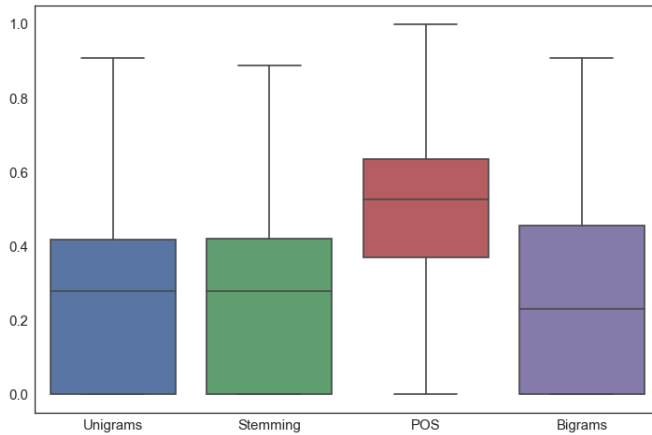


Figure 4.5: F_1 score comparison of the different NLP techniques used.

In Figure 4.6, we can observe the behaviour of the different resampling techniques used during the experiments. ROS is the technique that shows more stability, although in some experiments it can also obtain very low results. The other over-sampling techniques (ADASYN and SMOTE) have a similar behaviour, but with a low performance. Finally, undersampling techniques show the worst performance in general. However, it is important to focus on particular cases because they can obtain the highest performance, as in the case of Tomek Links in combination with POS and Logistic Regression.

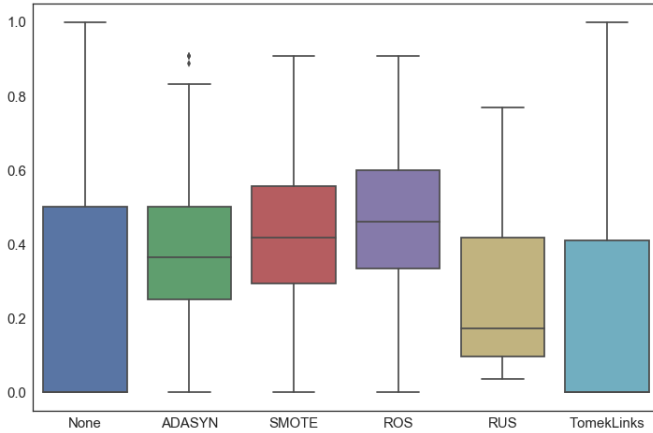


Figure 4.6: F_1 score comparison of the different NLP techniques used.

Finally, Figure 4.7 shows the performance of the different machine learning algorithms. SVM and Logistic regression present the best performance in general. In the case of Logistic Regression, it was expected, after reviewing the previous results, that it could obtain good performance. On the other hand SVM, seems to have a good performance, but it never obtains an excellent result. Thus, we can say that SVM is a robust approach that should be studied deeper in future to discover if it can also obtain comparable results to Logistic Regression.

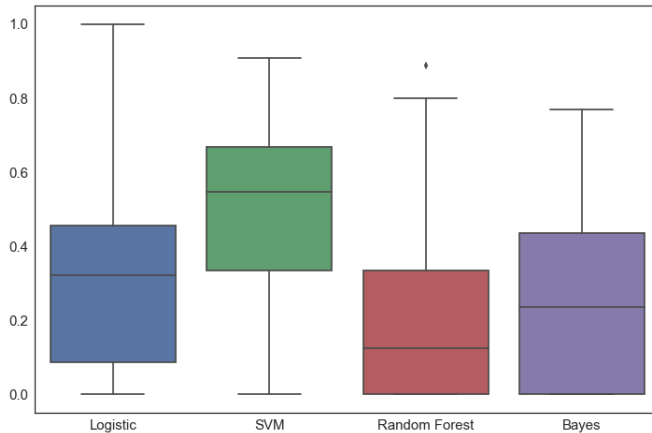


Figure 4.7: F_1 score comparison of the different machine learning algorithms evaluated.

4.4.2 Deep models Results

As with the shallow models, hyperparameters including learning rate, hidden layer size and number of layers can strongly affect model performance. In our preliminary study, the rule classifier is fine-tuned using training and development sets while the test set is kept totally untouched for reporting the system performance. Table 4.8 summarises the chosen hyperparameters for the experiments. These hyperparameters for our models were tuned on the development set by grid search. We experiment by tuning the hyperparameters with different settings: learning rates

(0.1, 0.5), LSTM layer sizes (32, 64). We train the networks architecture with the backpropagation algorithm to update the parameters for each training example with AdaDelta (Ruder, 2017) with a batch size of 16. Our epoch size is set to 60. From the training set of sentences, 10% of the sentences are held out as validation set. This allows us to evaluate the model in the training phase by determining the best F_1 score for early stopping (Caruana et al., 2001). If there is no improvement in the F_1 score within the last five consecutive epochs, the systems performs an early stopping. For the implementation of the neural networks, we use Keras 2.1.6 ² (Chollet, 2017) with Tensorflow ³ in the backend. Keras is becoming the ‘Lingua Franca’ of DL while Tensorflow has the largest active community by far.

Table 4.8: Selected LSTM Hyperparameters after cross validation.

Parameter	Value
Word Emb. Size	300
LSTM layer size	64
Batch Size	16
Epochs	60
Learning Rate	0.1
Kernel Initializer	Glorot

Experimental results in Table 4.9 show that the “Max-Mean” model with BLSTM and the use of transfer learning through

²<https://keras.io/>

³<https://www.tensorflow.org/>

Table 4.9: Summary of the deep networks architectures with best binary and macro F_1 score.

Transfer Learning	Agg. Mode	Bidirectional	Macro F_1	Prohibition F_1
True	Max-Mean	True	92.77	85.71
True	No	True	91.53	83.33
True	No	False	89.15	78.57
True	Max-Mean	False	87.96	76.19
False	No	False	78.30	57.14

word embeddings achieves the best performance with a macro F_1 of 92.77% and a binary F_1 of 85.71%. In the table, we also include the best result without using word embeddings; and it can be observed that performance is reduced from 89.15% to 78.30% in macro F_1 (improvement of 13.91%) while from 78.57% to 57.14% in prohibition F_1 score (an improvement of 36.84%).

4.5 Discussion

As there is the possibility of introducing serious health-related risks due to the provision of any wrong rule, it is critical that the phytosanitary rule category classifier provides information to the FMIS with the maximum possible accuracy. The best approach found in our experiments is a rule classifier with the combination of POS tagging, Tomek Links and Logistic regres-

sion. It obtains a promising performance of 68.8% in F_1 score with 84.46% precision and 60% recall. On the other hand, although, deep model experiments are preliminary, they have shown a really promising performance. They have surpassed shallow models especially when word embeddings are used. Regarding the binary F_1 , the improvement is of 24.57%; this result will be deeply studied and extended in future work. Although the ideal result would be 100%, this is unrealistic and in literature no real automatic system can achieve it. A human annotator/software developer could accomplish a closer performance, but due to the regulations overload, it would be difficult to have all the information that an automatic system could process. In addition, with the automatic extraction of rules, the information provided by the FMIS could rarely be outdated. Moreover, as Nash et al., 2011 also mentioned, until new algorithms and approaches will be researched, the original text of the rule must always be provided to the farmer.

In addition, this approach could be used as a computer-aided tool that human annotators could use to translate regulations into a formal semantic representation that could be executed within the FMIS. Therefore, this system could be seen as a component of semi-automatic rule extraction where the automatic part will increase its role with inputs from future NLP/embeddings, data augmentation techniques and ML advances. On the other hand, although there are multiple language constructs for each sentence type, these are limited. Maybe, some heuristics or post-processing could improve the performance. However, in a research context, it is preferable to use only the techniques as they are in order to observe their current

possibilities to automate the rule translation and avoid ad-hoc informal solutions as much as possible.

Finally, this chapter agrees with Nash et al., 2011 that obligation and prohibition are a good starting point for transforming rules in a machine-readable format (Step 1 of their framework); and next steps should extract the fine-grained constraints that are contained within the rules and that represents the actions that are obligated or prohibited.

4.6 Summary

In this chapter, we have evaluated whether it is possible to use ML techniques in combination with NLP and resampling techniques to classify rules between prohibitions and obligations and, consequently, the applicability of these techniques to implement a module that can be integrated within a FMIS that supports decision making based on regulations and production standards. To the best of my knowledge, this is a first attempt to combine different automatic rule classification approaches in the agriculture domain. The best approach found in our experiments is the combination of POS tagging, Tomek Links and Logistic regression. This combination obtains a F_1 score of 68.8% with a precision of 84.46% and a recall of 60%, which is a promising result that will be improved with the advances in ML and NLP research. The rule classifier obtained can be used as a computer-aided tool that human annotators can use to translate regulations into a formal language that could be executed within the FMIS. What can be concluded observing these res-

ults it that the algorithms are all data-dependent, meaning that their performance will heavily depend on the geometry of the dataset they are trained on. SVM may outperform logistic regression on one dataset, and produce lower results on another one.

In addition, some preliminary results have been obtained using deep models. These have been really promising surpassing shallow models with a macro F_1 of 92.77% and a binary F_1 of 85.71%; what translates into an improvement of 24.57% with respect to shallow models.

Moreover, introducing new techniques of information extraction, the spatio-temporal constraints could be automatically extracted and integrated within the FMIS. Therefore, an end-to-end system would be operative and regulations written in natural language will be automatically translated into machine-readable formats.

Chapter 5

Rule Constraints Extraction

In science there is and will remain a Platonic element which could not be taken away without ruining it. Among the infinite diversity of singular phenomena science can only look for invariants.

Jacques Monod
Chance and Necessity

5.1 Introduction

In this chapter, we add a new component to the rule extraction system. Our goal is to find a suitable DL architecture for building an end-to-end constraints extraction system in the rules previously categorised by the rule classifier. In Figure 5.1, it is

contextualised the rule constraints extraction inside the whole development of the rule extraction system that is proposed in this thesis. We evaluate different state-of-the-art neural network architectures to label the meaningful parts of textual rules (i.e., the restrictions) found in a phytosanitary products registry (phenological stages, maximum number of applications, temporal relations, etc.).

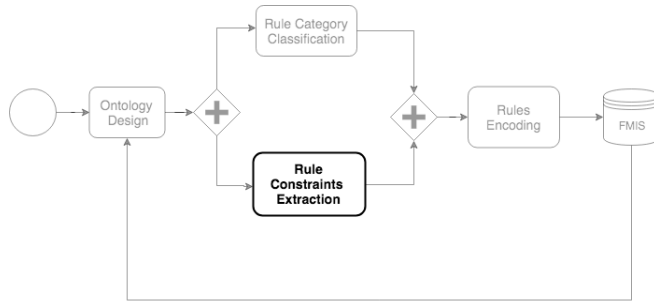


Figure 5.1: Rule Constraints Extraction: Contextualization of this chapter in the whole rule extraction workflow.

Entities and events are central objects of languages. Therefore, it is critical for computers to recognise such entities so that they can understand human languages. This is essentially the target of IE, an active branch of research in NLP since last decades, which aims for identifying entities, events and the inter-connections within the text. The ultimate goal is to transfer information in text into a more accessible format for other

computer applications such as question answering, information retrieval and knowledge base population among others. Entity mentions are continuous sequences of words in the sentences that mention some entities in the real world. Entity mentions can appear in various forms, including names, pronouns (i.e, “it”, “its”, “which”, etc.), and nominals (i.e, nouns, noun phrases, etc). In our case, the IE system should be able to recognise for example “flowering”, “fruiting” as phenological stages, “Greenhouse” as a treatment site, “3” as the maximum number of treatment applications in a season and “1” as the maximum number of treatment applications in an specific interval. A variety of methods proposed for IE mainly fall into three categories:

1. Dictionary-based methods that utilise lists of terms from diverse resources to identify entities in texts;
2. Ontology-based methods that map each unit of a text into one or more domain-specific concepts; and
3. ML methods that build models based on labeled corpora to identify entities.

Among them, ML methods are usually superior to the other alternatives because of their good performances and robustness when a large labeled corpus is available (Segura-Bedmar et al., 2013; Krallinger et al., 2015). As stated in the previous chapter, before the emergence of DL, the traditional pipeline to automatically extract information have employed different tasks which usually has two main steps:

1. Construct Linguist features: A large portion of the previous research effort in IE has been spent on developing feature pipelines that involves various NLP modules and resources to extract different effective linguistic feature sets for different subtasks. For instance: orthographic features, parts of speech, semantic features and morphological characteristics (e.g., prefixes and suffixes) among others. However, despite much effort on hand-designing feature representations for IE, the resulting feature sets might not be necessarily optimal. Feature engineering is a manual, time-consuming and expensive process that requires much linguistic intuition as well as domain expertise. Some researchers, such as Nguyen, 2017, mainly envision the following problems with the feature-based approach:
 - (a) Feature incompleteness: since domain knowledge is often incomplete, the feature engineering process might finalise without some important features.
 - (b) Feature redundancy: detecting interactions among the engineered features is challenging, potentially causing noise and redundancy in the feature sets.
 - (c) NLP software bugs: the NLP modules for feature extraction might involve errors, leading to noise in the features they generate and impairing the final performance of the system.
 - (d) Data sparsity or unseen word problem: the feature-based usually miss some relevant words during training and the ML models may do not take into account

them in the test data causing an unsuitable performance.

- (e) The semantic features are not easy to obtain: they require a large number of domain experts to build large-scale dictionaries.
- (f) The semantic features are hard to update: Domain knowledge is dynamic due to the real world challenges, and it is required that domain experts update knowledge resources with the latest advances.

2. Model Training: Once the linguistic features have been extracted, they are used to feed ML algorithms. The main traditional models for IE through sequence labelling are Hidden Markov Models (HMM), Maximum entropy Markov models (McCallum et al., 2000) and Conditional Random Fields (CRF) (Lafferty et al., 2001), which achieve relatively good performance (Ratinov and Roth, 2009; Passos et al., 2014; Luo et al., 2015). For some authors, CRF is one of the most reliable sequence labelling methods, since it has shown good performances on different kinds of tasks. SVM was also used in some works such as Zhang, 2004. However, some of these traditional linear statistical models heavily rely on hand-engineered features which, as explained previously, are difficult to collect and define besides costly to develop. For this reason, in recent years, researchers have been eager to use DL models, which can learn the feature representation from the raw input automatically. They have shown to be promising techniques

for NER tasks due to their ability to learn from the context surrounding the words in a sequence (Lipton et al., 2015). Among the different architectures, BLSTM has increasingly been employed for IE, yielding state-of-the-art performance in works such as Ma and Hovy, 2016; Wei et al., 2016; Habibi et al., 2017 and Luo et al., 2018. Sometimes, BLSTM is used to learn optimal contextual vector representations of every linguistic unit in a sentence to be taken as input to a the CRF. Finally, some models also take as input character-level embeddings of words to their BLSTM models, bringing the further outperformance to their IE models (Habibi et al., 2017; Luo et al., 2018).

It is important to remark that the IE task is full of challenges due to the following reasons: (1) the limited number of supervised training data; (2) new agricultural entity names are increasing constantly; (3) the authors of regulations do not always follow proposed standardised rules or formats. (4) Some information chunks contain less abstract information chunk, and this can be tagged as the simple information label. Maybe for these reasons, nowadays, there are very few works in literature that focus on IE from agricultural regulations. For example, Patil et al., 2013 used unsupervised learning to extract crops, diseases and chemical treatments. Otherwise, Malarkodi et al., 2016 proposed an approach for labelling crops, chemicals, locations and temperatures among others by using CRF. None of these examples have used DL techniques.

5.2 Deep NN Architectures for IE

In this section, we discuss widely the recent deep neural networks methods that are able to accomplish IE without requiring hand-crafted features. Our contribution is to show an end-to-end methodology to automatically label/tag meaningful parts of the phytosanitary regulations using a DL model. A model that does not require task-specific resources, feature engineering, or data preprocessing beyond pre-trained word embeddings on unlabelled corpora. Thus, our approach can be applied to a wide range of sequence labelling tasks on diverse agricultural regulations of different countries. We have used architectures developed in the state-of-the-art literature (dos Santos et al., 2015; Ma and Hovy, 2016; Lample et al., 2016; Strubell, 2017; Liu et al., 2017). Moreover, we have followed the framework proposed by (Yang et al., 2018), and we study three main neural components: (i) character sequence representations; (ii) word sequence representations; and (iii) inference layer. An example is shown in Figure 5.2. Green, red, yellow and blue circles represent character embeddings, word embeddings, character sequence representations and word sequence representations, respectively.

Specifically, we explore three neural model design decisions: character sequence representations, word sequence representations and inference layer. The input vector of the architecture is the concatenation of word embedding and character-level embedding from a sentence and feed them into a BLSTM or CNN to model context information of each word. Thereafter, on top of BLSTM or CNN, we compare two different inference layers:

a CRF to jointly decode labels for the whole sentence and a Softmax layer that makes a local decision without taking into account the label context.

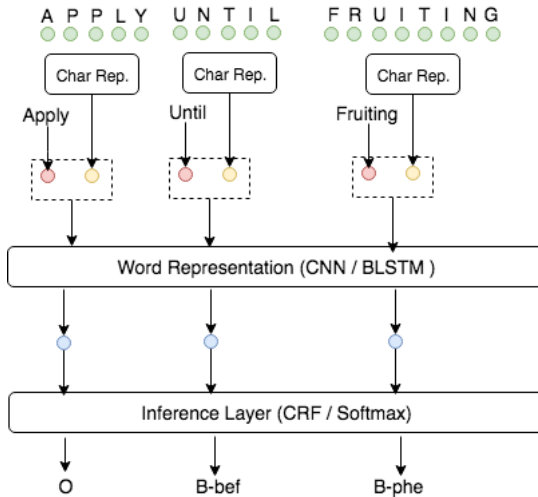


Figure 5.2: Neural sequence labelling architecture for sentence “Apply until flowering”.

Our model considers the regulations as a set of classified sentences where each individual sentence in turn consists of a sequence of words. The output of the architecture is a sequence of labels. Outputs could be transformed easily in the form of formalised sets of rules (maybe within a spreadsheet format or

a Java program). In this section, we describe the layers of the neural network architectures evaluated in this research.

Word Modelling

1. Word embeddings: As explained in previous chapter, word embeddings replace the hard matches of words in the feature-based approach with the soft matches of continuous word vectors. Afterwards, the multiple layers of hidden vectors further abstracts the word embeddings to automatically obtain underlying feature representations from data. Consequently, word embeddings mitigate the unseen word problem while the whole DL models help to avoid feature engineering providing effective feature representations. We also use the same pre-trained word embeddings as the previous chapter, and therefore, statistics can be seen in Table 4.2. We consider that it is worth investigating the effects of unsupervised semantic features based on word embeddings on DL systems. As it was observed in the rule category classifier development, it can be hypothesised that word embeddings should improve the performance of the IE system. Specifically, the recall is expected to improve because for the rare words in the training and test sets, the word embeddings induced on a large-scale unlabelled corpus naturally perform smoothing as they are dense, real-valued vectors.
2. Character sequence representation: The sole dependence on word embeddings will ignore explicit character level features like the prefix and suffix; and character embeddings

incorporate more morphological information on character level, which can not be included in word embeddings. On the other hand, character features such as capitalisation can be represented with character embeddings through neural networks without human-defined features (Lample et al., 2016; Ma and Hovy, 2016). They have been proven to be crucial for successful sequence labelling tasks.

In summary, two kinds of automatically constructed word representations are used in this chapter: (i) word embedding, which is trained from a large amount of text, and (ii) character-based representation randomly initialised, which can capture orthographic features of words.

Word Sequence Representation In the DL models for IE, layers of hidden vectors are put on top of word embeddings in order to capture hidden syntactic and semantic properties; in other words, to capture the context of words. In this chapter, we use the BLSTM instead of a single forward network. Each BLSTM has two separate hidden layers: forwards and backwards which are used to capture past and future information individually. The two hidden layers are concatenated to form the final output. For instance, given a sentence (x_1, x_2, \dots, x_n) , for each word x_i , we apply LSTM to compute the representation l_i of left context for the sentence, and, then we can get representation r_i of the right context by reversing the sentence. Concatenation of the left and right context representations gives the final representations $[l_i, r_i]$ and this representation is very useful for the IE system.

We also use CNNs, which combine diverse architectural ideas to extract features horizontally from multiple words allowing the network to extract higher level writing style. The kernel size in the convolutional layer defines the number of words to consider, providing a grouping parameter (Collobert et al., 2011; dos Santos et al., 2015; Strubell, 2017).

Inference Layer The inference layer, which actually is the tagger, takes the extracted word sequence representations as features and assigns labels to the word sequence. A very simple but effective labelling model is to use the hidden layer ($[l_i, r_i]$) as features to make independent labelling decisions for each output (y_t) by using a Softmax layer (Ling et al., 2015). However, despite the success of Softmax in simple problems like Part-Of-Speech tagging (Collobert et al., 2011), the assumption of independence of output labels limits its application in other common NLP tasks where there are strong dependencies across output labels (e.g., named entity recognitions, semantic labelling, etc.) (Huang et al., 2015; Yang et al., 2018). In other words, in this kind of tags, a “grammar” that captures the correlations between adjacent labels imposes constraints impossible to model with Softmax (even when BLSTM is used). Thus, the IE system fails to model by the independent decisions. For example, in our case, the tag “I-PHE” cannot follow behind the tag “B-AFT”. For this reason, statistical linear models such as CRF can be used. In summary, Softmax and CRF have their own advantages and disadvantages. Softmax is better for modelling long sequences of words, but the label for each word is predicted

independently and not as a part of the sequence. CRF is better for modelling the entire sequence jointly, but need hand crafted features to obtain significantly good results. In this thesis, we compare the use of Softmax and CRF as inference layer implementation. This comparison has also been explored in sequence tagging literature (Huang et al., 2015).

Dropout In this thesis, we also study if dropout can be beneficial to the IE task, which can avoid the over-fitting problem. Therefore, we apply dropout on the weight vectors directly to the final layer after the BLSTM. We fix dropout rate at 0.2.

5.3 PCT-O Extension

As MATTER methodology explains, during the annotation process, new concepts and definitions can arise in the domain ontology. In this case, the PCT-O model presented in Chapter 3 needs to be extended with new some new concepts and attributes. The main modifications affects the class *ChemicalTreatment*, which has been extended with two additional attributes: *maxApplications* and *site*. These attributes represent, respectively, the maximum number of applications that a chemical treatment can be applied during a season; and the site where the treatment can be applied (i.e.: outdoors and greenhouse). In addition, the *ChemicalTreatment* is a composed by different intervention periods (*InterventionPeriod* class) with a maximum number of applications during these periods and a minimum number of days between applications. Finally, these interven-

tion periods are composed by different intervals (*Intervals* class) that represent constraints with the phenological stages of the crop when a chemical treatment can be applied.

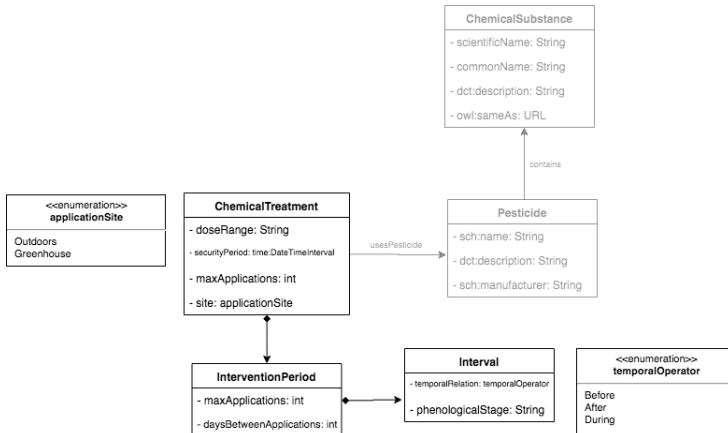


Figure 5.3: Extension of PCT-O in order to represent the complexity of a chemical treatment application.

5.4 Experiments

This section describes the experiments performed on end-to-end neural networks. Although the process presented in this work, is suitable for labelling information from phytosanitary regulations in different countries, we present the Spanish case as an

example of use. In addition, we investigate the main influencing factors to system performance, including the character sequence representations, word sequence representations and inference algorithm.

5.4.1 Constraints in Spanish agricultural standards

In the same way that happened with the rule category classifier, in order to create a sequence labeller (the rule constraints extractor) with DL, the labeller must be previously trained with a gold corpus, which is a set of manually annotated texts that serves as a basis for the training and evaluation. In this work, we use a corpus focused on the official information about crops and authorised pesticides from the Spanish phytosanitary products registry. As explained in the previous chapter, this repository stores 2,426 PDFs that contain authorisations and instructions about how to apply phytosanitary products to comply with environmental regulations. In addition, in order to have the last regulations, we have developed an automatic process that checks periodically if any regulations have been updated. We have each the of rules linked to its specific phytosanitary product; and each product has its own unique identifier. Therefore, if we detect that in the official Spanish phytosanitary products registry, a product document has been updated with new rules, we remove the previous rules and the whole rules are extracted again with the techniques shown in this chapter. It is also important to highlight that during the corpus development we have found some inconsistencies among different documents. For ex-

ample, the same phenological stage could be found to have diverse BBCH codes ¹ assigned.

To address the annotation task, we have converted the attributes and classes of the PCT-O extended version in a set of label types shown in Table 5.1. Moreover, there are two label types (“xor” and “rep”) that do not appear in the Table 5.1, but they are necessary to capture the syntactic complexity of the rule accurately; “xor” is used when the treatment has two ways of being applied, and “rep” is used when the treatment has assorted intervention periods. Another terms (e.g., apply, do, never) are not labeled because they do not represent restrictions, but linguistic signs of the rule category (prohibition, obligation), which were studied in the previous chapter. Since some concepts are expressed through multi-word expressions (continuous sequences of tokens), the “IOB2” annotation scheme (standard mentioned in CONLL 2003 shared task) is used (“B” indicates the beginning of an event, “I” is for inside an event, and “O” is for outside (the word do not refer to an event) (a.k.a no-entity tag).

¹<https://bit.ly/2InANNY>

Table 5.1: Label types used for annotation.

Label	Definition
phe	The available phenological stages for the treatment
dur	Temporal relation: During
aft	Temporal relation: After
bef	Temporal relation: Before
mac	Number of times a treatment can be applied during a season
mai	Number of times a treatment can be applied during an intervention period
pla	Place where the treatment can be applied

It is important to note that there are another different representations for sequence labelling that we have not used. For example:

- IO: Here, only the I and O labels are used. We have not used this option because we cannot distinguish between adjacent chunks of the same named entity.
- IOB1: Here, I is a token inside a chunk, O is a token outside a chunk and B is the beginning of chunk immediately following another chunk of the same entity. IOB2, used in this research, is same as IOB1, except that a B tag is given for every token, which exists at the beginning of the chunk.

- IOE1: An E tag used to mark the last token of a chunk immediately preceding another chunk of the same named entity. IOE2 is same as IOE1, except that an E tag is given for every token, which exists at the end of the chunk.
- START/END: This consists of the tags B, E, I, S or O where S is used to represent a chunk chunk containing a single token. Chunks of length greater than or equal to two always start with the B tag and end with the E tag.

An example of tagged rule can be found in Figure 5.4, from the part of a sentence “Never|apply|before|10|leaves”, the corresponding label is “O|O|B-BEF|B-PHE|I-PHE”. The corpus statistics are shown in Table 5.2 and Table 5.3. It is important to remark that the use of a large corpus is especially relevant in some tasks (e.g., image classification, object detection) because automatic feature extraction can involve millions of features. However, if feature diversity is not so large (our corpus contains 5,459 words), suitable models that extract meaningful patterns from data can be developed, by using techniques such as pre-trained word embeddings (Joulin et al., 2017). The gold corpus developed is publicly accessible ².

²<https://bit.ly/2G0KkZF>

1. Never apply before 10 leaves
B-bef B-phe I-phe
2. Apply only until fruting
B-bef B-phe
3. Apply in greenhouse 3 times or 4 times outdoor
B-pla B-mac B-xor B-mac B-pla

Figure 5.4: Example of labeled rules.

Table 5.2: Gold Corpus Statistics.

Corpus Statistics	
No. of rules	273
Rule length average	22 words
No. of labels	12
No. of entities	1803
No. of words	5459
No. of unique words	679

Table 5.3: Number of Labels in the corpus.

Label	#
B-phe / I-phe	610 / 1031
B-dur	252
B-aft / I-aft	213 / 63
B-bef	203
B-mac	129
B-mai	18
B-pla / I-pla	16 / 9
B-xor	59
B-rep	27

This process is practically language-agnostic: in the case that this experiment was reimplemented with another language,

only the word embeddings layer should be replaced; and only in the case that a pre-trained embedding is used. The rest of the process could be reused in the same way.

5.4.2 Preprocessing

The document preprocessing stage was developed and explained in the previous chapter. It is valuable to highlight that our deep model operate on the tokenized sentences. We did not make any restrictions on the sentence length. Rather, we used the maximum length of the sentences in a batch. All shorter sentences in that batch are padded with a mask (“0 padding”). In our experiment, apart from replacing all digits with zero, we did not do any additional preprocessing on the gold corpus.

5.4.3 Hyperparameter tuning

As explained in the rule category classifier training process, hyperparameters including learning rate, hidden layer size and number of layers can strongly affect model performance. In our study, the IE is fine-tuned using training and development sets while the test set is kept totally untouched for reporting the system performance. Table 5.4 summarises the chosen hyperparameters for our experiments. Due to temporal constraints, the hyperparameters for our models were tuned on the development set by random search (contrary to the grid approach used in the previous chapter). We experiment by tuning the hyperparameters with different settings: learning rates (0.1, 0.2, 0.5), LSTM layer sizes (50, 100, 150) and CNN layer sizes (32, 64,

128). We train out networks architecture with the *backpropagation* algorithm to update the parameters for each training example with stochastic gradient descent (SGD) with batch size 1 and a fixed learning rate. SGD is a variant of gradient descent. Instead of performing computations on the whole dataset, SGD only computes on a small subset or random selection of data examples. We explore other more sophisticated optimisation algorithms such as the adaptive ones, RMSProp and Adam (Ruder, 2017), but in preliminary experiments they did not improve upon plain SGD. After some empirical tests with different sizes (20, 40 and 55), we have set our epoch size to 55. In each epoch, we divide all the training data into batches, then process one batch at a time. In each batch, we firstly get the output scores from the BLSTM for all labels. Then we put the output scores into CRF model, and we can get the gradient of outputs and the state transition edges. From this, we can back propagate the error from output to input, which contains the backward propagation for bi-directional states of LSTM. From the training set of sentences, 10% of the sentences are held out as validation set. This allows us to evaluate the model in the training phase by determining the best F_1 score for early stopping (Caruana et al., 2001). If there is no improvement in the F_1 score within the last five consecutive epochs, the systems performs an early stopping. Pre-trained word embeddings are evaluated with fine-tuning. As in the previous chapter and due to the ease of development, for the implementation of the neural networks, we use again Keras 2.1.6 with Tensorflow.

Table 5.4: Hyperparameters

Parameter	Value
Char Emb Size	10
Word Emb Size	300
CNN window	3
CNN layer size	32
LSTM layer size	50
Batch Size	1
Epochs	55
Learning Rate	0.1

5.4.4 Evaluation

As in the previous chapter, F_1 score (Eq. 4.3) is used as the evaluation metric for sequence labelling, where precision is the ratio of correct labels in the sequence labeller output and recall is the ratio of the correct labels in the gold corpus. This evaluation technique measures the correspondence between the labels that the sequence labeller generates and those of the gold corpus. To compare the overall performance among neural architectures, we use the micro-average approach because in a multi-class classification setup, this approach is preferable if there is class imbalance (See Table 5.3 for more details). To reduce the volatility of the system, we conduct each experiment 10 times under different random seeds, and report the mean for each neural architecture. The evaluation is performed under this two criteria:

1. Word-level evaluation: Here we do not consider that the

whole entity is completely tagged, but only the different tags that are found in the gold corpus. At this level, it is also presented a comparison of the performance of each of the main components of the neural architectures. This evaluation can clarify which are the most promising future research directions.

2. Rule-level evaluation: In the rule extraction problem, it would be interesting to know if it is possible to extract all the constraints (i.e. all the entities) inside a rule. Thus, it is also assessed the proportion of rules that are completely and accurately tagged. This comparison will be done between an ensemble of taggers and the best overall performer one.

5.5 Results

This section shows the results of the experiment performed with the different neural architectures. To represent the neural network architectures, we use the structure “character representation-word representation-inference layer”. Moreover, to simplify the description, we use the following nomenclature: “N” and “C” to represent No char and character embedding representation in the character representation layer; “B” and “C” to represent BLSTM and CNN structure in the word representation layer; and finally, “C” and “S” to represent CRF and Softmax layers in the inference layer. This can be seen in Table 5.5.

Table 5.5: Nomenclature used to describe neural architectures in table 5.6.

	BLSTM		CNN	
	Softmax	CRF	Softmax	CRF
Char	C-B-S	C-B-C	C-C-S	C-C-C
No Char	N-B-S	N-B-C	N-C-S	N-C-C

5.5.1 Word-Level evaluation

Examining the results in Table 5.6, we think that there are some relevant facts to remark. Each label has its own best sequence labeller (in bold format), so we can infer that in our gold corpus, a best algorithm for the complete translation of human-oriented regulations into computer-oriented regulations does not exist and an ensemble of neural network architectures is necessary to label the rules with the highest performance. This will be observed in the section 5.5.2. Another important observation is that a complex architecture such as the C-B-C obtains 0% F_1 score within 4 different label types. In other words, this architecture cannot model the patterns that other simpler architectures can. The main reason is that these labels have a small representation in the corpus and the complexity of the architecture is an impediment to obtain a good performance. Related to this, there are two labels (“B-mai” and “B-rep”), with which all the architectures obtain low performances. There are two main reasons for this result: firstly, as shown in Table 5.3, these labels contain few examples and DL approaches may have

difficulties to extract meaningful patterns. Secondly, the words annotated by these labels present polysemy (i.e.: the same word can be labeled differently), making the labelling more tricky. Finally, in the last table row, we show the micro-average F_1 score for each neural network architecture.

Table 5.6: Architecture’s F_1 score per label type.

Neural Architectures								
Label	N-B-S	N-C-S	C-B-S	C-C-S	N-B-C	N-C-C	C-B-C	C-C-C
B-phe	90.96	85.38	90.08	84.74	90.24	88.74	90.47	88.70
I-phe	87.50	79.63	89.99	79.68	89.37	86.43	88.82	87.62
B-dur	88.44	86.73	92.07	86.17	87.00	89.12	89.11	89.90
B-bef	96.26	94.62	95.95	94.67	93.75	89.73	92.61	92.14
B-aft	89.89	93.74	92.07	94.58	93.26	92.85	91.17	93.20
I-aft	66.66	75.00	74.50	61.22	72.33	72.41	67.80	74.57
B-mac	78.78	79.53	74.86	83.42	75.00	79.24	74.07	84.39
B-xor	82.92	82.92	87.80	82.34	79.99	82.34	66.66	79.99
B-mai	33.33	34.78	13.32	42.10	12.49	29.62	0.0	43.47
B-rep	58.82	38.71	56.25	45.71	44.44	48.78	0.0	51.42
B-pla	95.65	95.23	90.91	100	80.00	90.91	0.0	81.82
I-pla	92.30	100	100	100	92.30	100	0.0	92.30
μ -Avg	87.18	82.70	88.30	82.84	87.26	85.73	85.81	87.01

Taking into account this result, the C-B-S architecture shows the highest performance and it can be considered as the best overall approach. This architecture will be used as a baseline

against the architectural ensemble. It is also remarkable that all the architectures with CRF in the inference layer, except C-C-C, do not obtain the highest performance in any label types. This seems to contradict the general belief that CRF is always a good approach to model sequences. This will be studied deeply in the next subsection. Finally, it is important to highlight that all the neural networks evaluated obtain performances over 82%, which are results quite akin to those obtained in the agricultural community sequence labelling benchmarks (Malarkodi et al., 2016; Patil et al., 2013). Again, it is important to remark that comparisons among literature works cannot be easily done, because datasets and approaches are divergent.

Following the framework proposed by Yang et al., 2018, the aim of this phase of the evaluation is the comparison of the different neural layers in order to study which ones lead to an overall better performance and therefore it is demonstrated its effectiveness for this IE in this dataset. In order to confirm that the differences are not due to chance, we have computed statistical Welch's t-test (Welch, 1951) with a confidence level of 0.1.

Char vs No char In our experiments, according to Figure 5.5 a), character information slightly improves the sequence labelling models. Moreover, the difference is statistically significant ($p < 0.1$).

CNN vs BLSTM In Figure 5.5 b), we can observe that BLSTM obtains a better performance than CNN. However, the

difference is not statistically significant ($p > 0.1$). From these results, we cannot conclude that the global word context information is necessary for sequence labelling.

CRF vs Softmax According to Figure 5.5 c), models with CRF inference layer do not outperform the models with Softmax layer under all configurations, proving that label dependency information is not effective in our corpus. Moreover, the difference is not statistically significant ($p > 0.1$).

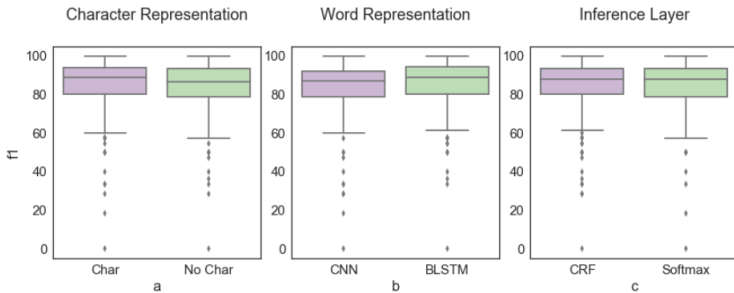


Figure 5.5: Comparison of layers performance in deep models.

5.5.2 Rule-level evaluation

As explained previously, it is also evaluated if by using the best tagger for each of the label types (“Ensemble”) (see Table 5.6), the proportion of rules that are fully labelled is increased when comparing to the best overall tagger (“C-B-S”). In table 5.7, it

can be seen that the tagger composed by the ensemble approach achieves a best performance. Specifically, it improves by 5.48 % the number of rules fully labelled by “C-B-S”.

Table 5.7: Comparison of the IE systems at rule-level.

Approach	% rules
C-B-S	41.73
Ensemble	43.29

5.6 Discussion

In this chapter, we have developed an end-to-end sequence labeller, which is a necessary step to automate the translation of human-oriented regulations into formal rules. Moreover, because deep neural networks automatically extract non-linguistic features, this approach can be applied to transform regulations in other countries. As a first step, we have extended the PCT-O conceptual model to represent the complexity of a chemical treatment applications, and we have translate this model into a set of labels. In our experiments, we have found that an architecture with character embeddings, BLSTM and Softmax (C-B-S) obtains the best performance. This system, with an overall performance of 88.3% F_1 score, overcomes the rest of the approaches. Currently, there is no benchmark for sequence labelling in the context of agricultural regulations and, therefore, it is tricky to directly compare our results with another works. However, there are two related works that are important to men-

tion. Patil et al., 2013 worked on the agriculture domain with 3 labels in contrast to our 12 labels. The highest precision obtained by the their algorithm is 66.2% for crop, 92.8% for disease and 88.6% for chemical. In other work, Malarkodi et al., 2016 extract crops, chemicals and locations among others, and obtain a precision of 83.24%, a recall of 83.13% and F_1 score of 83.18%. Therefore, it can be concluded that the results are quite similar to the state-of-the art works in the agricultural domain.

Moreover, it is evaluated if an ensemble of the best taggers can surpass the C-B-S architecture, which is the best overall solution. The results indicate that there is a difference of 5.48% in performance. Thus, if the number of rules that must be extracted is large, this ensemble of taggers could be a possible solution to improve the performance. However, it is evident that in spite of the fact that the performance word-level is promising, at rule-level the results must improve if an automated solution is the final aim. At this moment with our approach, a semi-automatic system would be the most realistic approach.

It is important to note that all the architectures presented in this chapter have only one hidden layer and maybe, they could be too simple to learn properly the linguistic phenomena. Another important observation is the low influence of the different neural layers by themselves. Different combinations accomplish better or worse performance, but we can conclude that none of the layers always improves decisively the performance of the sequence labeller. Character embeddings improves slightly the performance, but it must be used with specific layers such as BLSTM in order to consolidate the improvement. In related literature, this is not the case, but they use larger corpus. and we

can hypothesise that this could be a consequence of our corpus size. Moreover, the use of the IOB annotation scheme with a small gold corpus can hinder the learning of sequential patterns. More complex schemes could lead to a better performance.

Finally, the design of the PCT-O extension has no take into account the alignment with another ontologies, although in future it will be studied how to align the *Interval* concept with the standard *Time* ontology ³.

5.7 Summary

In this chapter, we have presented an empirical evaluation of 8 state-of-the-art deep learning architectures to develop and end-to-end sequence labeller for the phytosanitary regulations in agriculture domain. To the best of our knowledge, this work is the first attempt in sequence labelling of phytosanitary regulations by using a systematic comparison of different DL techniques. We have evaluated the performance of three main layers: a character sequence representation layer, a word sequence representation layer and an inference layer. For this evaluation, we have used a gold corpus based on the Spanish phytosanitary products registry. Moreover, an extension of the PCT-O developed in Chapter 3, has been presented. This model has been translated into labels that DL architectures had to learn to automatically extract. In our experiments, the best sequence labeller system has a character embedding layer as the character

³<https://www.w3.org/TR/owl-time/>

sequence representation, a BLSTM as the word sequence representation and a Softmax as the inference layer. This architecture achieves 88.3% F_1 score, which is comparable to results obtained in related work.

Despite the good results, we believe that the performance can be further improved. DL is often used in problems that have very large datasets with thousands or hundreds of thousands of instances. For this reason, in future work, we will evaluate techniques for increasing the corpus size (e.g., silver corpus). Moreover, the output labels could be annotated with BIOES standard, since this scheme has been reported to outperform others such as IOB (Yang et al., 2018). Finally, a multi-channel CNN for labelling will be evaluated. This architecture involves using multiple versions of the standard model with different sized kernels. This allows the document to be modelled at different n-grams (groups of words) at a time, whilst the model learns how to best integrate these interpretations. All these improvements bring closer the human-oriented regulations to computer-oriented regulations.

Chapter 6

Conclusions

Without music, life would
be a mistake.

Friedrich Nietzsche
Twilight of the Idols

6.1 Summary of Contributions

This work has aimed to develop a framework to automatically transform NL regulations into a set of formal rules. We have presented a refined methodology for extracting rules from agricultural regulatory text in order to assist ACC in the PA domain. Although the implementation presented during the chapters of the thesis face the steps from regulatory text to semantically annotated rules; the methodology proposed covers the whole rule extraction lifecycle from regulatory text to executable rules. It is based on the state-of-the art techniques and, as explained in the previous chapters, the results of the approach, such as the domain ontology model and the ML algorithms are open and extendable. Although each of the pre-

vious chapters explain the contributions deeper, a summary of the main contributions of this thesis is the following:

1. This thesis has presented a study of the current techniques to translate NL regulations into a set of formal rules. There is an amalgam of possible solutions for rule extraction with contrasting requirements and approaches. However, although some works try to provide a generic framework or methodology for rule extraction, it is not clear that these results are practical when applied to specific real-world case studies in the PA domain. In spite of the fact that the problem of extracting rules or conditions from legal texts is still open, some related methodologies have been highly inspiring. It is also important to remark that although related experimental results can be informative, the goal of these works and their datasets are different; and consequently an experimental comparison with the performances reported is not fulfilled.
2. This thesis has developed its own methodology to translate regulations into rules by adapting the MATTER methodology and the framework presented in Nash et al., 2011. We consider that the use of MATTER is important to standardise the rule extraction process and make the process and annotation products transparent, portable and extensible; as it has been mentioned in Section 1.4.
3. This thesis has designed a domain ontology (PCT-O) developed to support decision in crop treatment application. PCT-O provides relations between crops, pests, and treat-

ments and includes the description of the conditions of these relations. Therefore, in PCT-O, it is possible to specify the period of time when a pest is harmful, when it is needed to react, and the nature of the treatments. PCT-O model has been created by reutilising concepts from other ontologies such as NCBI and PubChem. In our proposed methodology, the domain ontology model is a key input to all the steps and its quality is one of the main factors that influence the rule extraction suitability for a real ACC system.

4. This thesis has evaluated the combination of diverse ML algorithms, NLP techniques and resampling methods for classifying texts rules between prohibitions and obligations. Empirical results have shown the importance of finding the correct combination of techniques because there are some alternatives whose performance is really bad, although they are quite similar to the best performer ones. Moreover, we have presented a preliminary study using DL techniques to classify rules with very promising results.
5. Finally, this thesis has presented a linguistic-agnostic end-to-end sequence labeller in order to automatically label meaningful information from regulations. This information contained in regulations is extremely fruitful in simplifying the space of possible actions or treatment prescriptions. For example, within a treatment prescription system, knowing the treatment application rules simplifies the problem of what farming operations are “legal”, and entire classes of illegal operations can be eliminated.

The DL algorithms have achieved very promising results, although improvements should be undertaken in order to extract the whole rules (i.e. all the constraints). Currently, this part of the process should be semi-automatic and complemented with human intervention.

6.2 Future Work

It is clear that there are still many challenges to overcome, and some problems still do not have suitable solutions. This indicates that, currently, tools for automatic rule extraction, rather than offering a definite answer, can at most provide a guess that will allow its researchers to take some action in a timely manner. As technology evolves, some of those limitations may be overcome, but there is still much to be investigated. Moreover, it is important to note that, in spite of the amount of experiments delivered in this thesis, the evaluation presented up until now has largely been of a theoretical/laboratory nature, which has ignored many of the practical problems associated with real farming operations. In this section, we divide the future work between tasks that will be improved (Section 6.2.1) and tasks that were out of the scope of this thesis but they are necessary to complete the whole process of rule extraction (Section 6.2.2).

6.2.1 Revision

Figure 6.1 contextualises the stage of the workflow that will be refined and improved in future work.

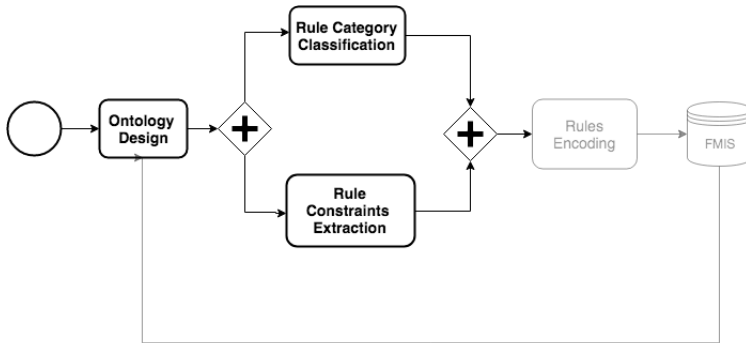


Figure 6.1: Rule Extraction workflow parts to be refined and improved in future work.

6.2.1.1 Ontology design: PCT-O extension and refinement

To reflect the complexity of crop treatment phenomena, new concepts, and the refinement of the existing ones, will be necessary. For example, the specialization of the *Pest* and *Pesticide* concepts will be some of the most relevant changes to be tackled. New concepts such as *Fungus*, *Insect*, *Weeds* or *Bacteria* alongside *Fungicide*, *Insecticide*, *Herbicide* and *Bactericide* will be added. Of course, the addition will be accompanied of new data sources and new ontologies to be studied.

Another area of future work is to integrate treatments adopted by other countries for the same illnesses/pests in the population of the ontology. This will allow complementing the pest

descriptions and comparing the approved treatments to detect differences between regions. These differences may show gaps in country legislations, and allow identifying better solutions for a region than the currently approved ones.

Another interesting extension would be to include other aspects of the use of chemical substances in the land. For example, PubChem repository contains information about the hazards of the use of the chemical substances, such as “Very toxic to aquatic life with long lasting effects” (Fu et al., 2015). This information, merged with water flow, crops or protected species distribution maps can be useful to determine the areas where a product can be used, or suitable alternatives for areas that forbid it. A complementary source of this information is the EU - Pesticide Database (European Commission, 2005) that stores the list of substances approved in each European member state for their use as pesticides. Finally, the ontology could be extended to integrate more detailed information about crops and their varieties. For example, the Spanish Ministry of Agriculture provides a collection of descriptive sheets ¹ containing information about the different crop varieties used in Spain. This collection provides information about the growth conditions, performance and resistance of the diverse varieties of species. This could be used to recommend the best variety for a field given its climate and the distribution of the registered pests.

¹<https://bit.ly/2V5H4jQ>

6.2.1.2 Rule Classification and Constraints Extraction

Annotation As shown in this thesis, in order to train the ML algorithms, it is necessary to annotate the regulations. Currently, the manual annotation for creating the gold corpus has been developed by a single person. However, in order to develop an industrial solution, a community effort should be required. There are different steps such as creating annotation guidelines and select a group of experts to annotate the regulations. Once this is done, in order to assess how well an annotation task is defined, we will use Inter-Annotator Agreement (IAA) scores (Carletta, 1993) to show how individual annotators compare to one another. If an IAA score is high, that is an indication that the task is well defined and other annotators will be able to continue the work. This is typically defined using a statistical measure called a Kappa Statistic (J.A., 1960). Having a high IAA score does not necessarily mean the annotations are correct; it simply means the annotators are all interpreting the instructions consistently. Currently, there are some other possibilities besides experts' annotation. For instance, crowdsourcing is another approach that is being used in the community. With this approach, the task is broken down into a large number of smaller tasks, and a large number of annotators are asked to tag only a few examples each. One popular crowdsourcing platform is Amazon's Mechanical Turk ².

²<https://www.mturk.com/>

Preprocessing Since open data is the source of information in this thesis, the acquisition of information still proves a demanding task, since information is produced from heterogeneous sources not necessarily interrelated and collaborated. Currently, political authorities publish regulations in PDF format and therefore, some preprocessing problems can arise. PDF file is analogical (scanning of a printed document) and therefore an OCR process sensitive to errors is necessary and this limits the quality of the extracted content. Most of the extracted text contains minor errors due to bad recognition of some characters. In addition to this, the non-plain text parts of the documents (e.g., captions of photos or tabular information) are not correctly extracted due to technological limitations limiting the model richness. For this reason, some external efforts such as political ones will be necessary to improve the whole rule extraction process.

Natural Language Processing To fully understand the NL, algorithms need to take into account not only the literal meaning semantic provides, but understanding of what the text is trying to reach. This level is called pragmatic analysis which is only beginning to be introduced into the NLP techniques. In this line, unsupervised NLP and more specifically word embeddings, have emerged as a crucial technique to improve the extraction of semantics from text. Thus, it will also be necessary to review the appearance of new word embeddings algorithms such as BERT (Devlin et al., 2018) and ELMo (Peters et al., 2018). BERT is a model that provided a significant step towards handle language-based tasks. It uses attention transformers instead of BLSTM

to encode word context. Moreover, versions of the pre-trained model on massive datasets are available for download. On the other hand, ELMo is a model that generates embeddings for a word based on the context it appears by looking at the entire sentence. It uses a BLSTM trained on a specific task to be able to create those embeddings.

Deep models development In the idealised classical supervised classification paradigm, certain assumptions are implicit: it is assumed that training data and the new samples are drawn from the same distribution; that the populated domain ontology is static and that the costs of different kinds of misclassification are known accurately and static. In real application, however, these assumptions will often not hold. They may well be swamped by uncertainties arising from mismatches between the apparent problem and the real problem. Therefore, there are some issues that should be reviewed periodically. For example the design of the sample used for training will be updated to avoid population drift. Moreover, errors in class labels will also be detected through the acquisition of new annotators. The ontology will also be reviewed to confirm that the concepts and relations still represent agricultural domain. The training process will also be updated if misclassifications costs change.

Finally, it is also important to remark that both the machine learning and deep learning are research fields that are advancing rapidly with major increases in computational power; and, therefore, new algorithms and techniques such as Capsule Networks (Sabour et al., 2017) are arisen constantly to solve

current pitfalls. For example, there is a flaw in the essential design of CNNs because the convolution operator is represented by a weighted sum of lower layers and it is difficult to express the hierarchical relationships between local features. On the other hand, capsules (grouped neurons) consider the spatial relationships between entities and learn these relationships via dynamic routing utilising a nonlinear function called squashing. Dynamic routing determines the connection strength between lower-level and upper-level capsules through coupling coefficient that is utilised to measure the similarity between the vectors that predict the upper capsule and lower capsule. Capsule networks have been shown their validity in the domain of text (Park et al., 2018; Wang et al., 2018).

Another point to review is the arisen of new optimization methods. Optimization is the basis of any ML methods, and in this thesis, we have only experimented with the most popular ones (i.e., Adam and SGD). The emergence of new algorithms, such as Adabound, will require an evaluation of their capabilities. These new techniques employ dynamic bounds on learning rates in adaptive optimization algorithms, where the lower and upper bounds are initialized as zero and infinity respectively, and both smoothly converge to a constant final step size, achieving the best performance in most tests when compared to other optimizers, such as Adam and SGD, while maintaining fast training speeds and hyperparameter insensitivity.

6.2.2 New approaches

Figure 6.2 contextualises the workflow stage that was out of the scope of this thesis, but will be faced in future work since it is highly related to achieve ACC.

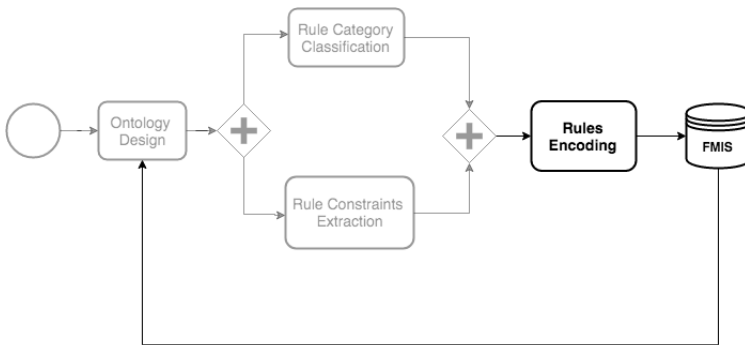


Figure 6.2: Rule Extraction workflow parts to be faced in future work.

6.2.2.1 Rule Normalization

Information Normalization There are some entities extracted by the rule constraints extraction system that can be expressed in diverse ways, but they refer to the same concept (i.e.: they are synonyms). For example, “Senescence” and “Beginning of dormancy”. Both of them are phenological stages, that although expressed with different terms, represent the same growth stage. To normalize these entities, in the future we will

develop a normalization step that will transform phenological stages names into the correspondent BBCH codes. The BBCH-scale is used to identify the phenological development stages of a range of crop species where analagous growth stages are given the same code.

Rules Encoding In order to execute rules or inferencing over extracted information at the FMIS level, it is necessary to encode the extracted information within a formal language. RuleML (Hirtle et al., 2006), SWRL (Horrocks et al., 2007) and RIF (W3C, 2009) are rule markup languages with different expressiveness, decidability and portability that could be used for rule normalisation and encoding. RuleML is an XML based markup language for the representation of rules, but without considerable consideration to features beyond representing and exchanging rules. SWRL is a combination of the languages OWL and RuleML. SWRL retains the full power of OWL DL but at small practical costs including decidability. SWRL bears resemblance to logic programming with Horn clauses and has a relatively human readable syntax in addition to the concrete XML syntax. A Horn Clause is one of the most restricted forms of FOL and is the most widely-used for logic-based inference-making. However, SWRL is not decidable and no implementation supports the full SWRL specification. RIF is an emerging W3C recommendation for the encoding and interchange of rules, but with few existing tools for it. It has also the expressive power of Horn clauses but without function symbols. It is important to remark that there are assorted types of formally-defined logic

with varying degrees of descriptive capabilities (propositional logic, predicate logic, modal logic, description logic, defeasible logic, etc.). A deep study will be made to find the most suitable for supporting automated reasoning for ACC.

6.2.2.2 Ruleset deployment into FMIS

There is no generic deployment scheme that fits every problem. Deciding what practices to use, and implementing them, is at the heart of what ML engineering is all about. Training algorithms is only one side of the coin, but there are a lot of interesting problems to solve on the infrastructure and deployment side.

We have faced preliminary research in order to find an FMIS where the rule extraction module can be incorporated. One of this FMIS is called Agroplanning³. Agroplanning is a modular cloud-based FMIS that treats the tractor as a centralized connected platform for data generation and reception. The aim of the system is to incorporate the tractor-centric approach defined by Fountas et al., 2014, and equipping agricultural service companies, farmers, cooperatives and machinery manufacturers with the tools to generate the first advanced precision farming services, improve efficiency and increase the precision of agricultural management.

As suggested by Nikkilä et al., 2012, another possibility would be to deploy the rule extraction system within a service oriented infrastructure. This is one of the most common type

³<https://www.agroplanning.com/es/my-agroplanning>

of ML workflow: a simple web service that takes in some parameters and spits out a prediction straight away; in this case, a formalised rule or a treatment prescription.

6.3 Final Conclusion

Besides the results presented in this thesis, I consider that the problem of extracting rules or conditions from legal texts is still open. However, it is also important to note that after the evaluation of some deep learning techniques, I consider that automatic or semi-automatic rule extraction is a very promising path to assist the regulatory compliance checking in a more effective and standardised way. This interest is due to the fact that the compliance checking task is currently tricky and extremely time consuming for humans; and, therefore, automating partially would become an important support. If effective rule extraction is achieved, we would obtain self-configuring FMIS to be able to make site-specific recommendations respecting up-to-date legal and voluntary restrictions on planned operations. This will lead to an improvement on crop protection management, a limitation of the uncontrolled acquisition of pesticides that leads to their overuse and misuse and a significant increase in both quality and quantity of agricultural products.

Although ACC is the ultimate aim, while some research and engineering problems persist, a initial solution could be making the standards accessible as a series of individual rules inside a user-friendly tool, which may be used by farmers to make more informed decisions according to legal requirements. Thus, the

farmer will ultimately have also access to the original official rule in a straightforward and comprehensible natural language way.

It is important to highlight that the ACC challenge is not only a technological one. This process requires the involvement of agricultural experts, of computer and data science experts, as well as advances in terms of organisational, ethical and legal arrangements. The move from publishing regulations as legal texts to publishing as individual rules in a machine-readable format is a large change in procedure which would require significant regulatory authorities activity. During the realisation of this thesis, it has been highlighted that some regulations are highly ambiguous, and the solution is an effort carried out by the publishers in order to clarify the rule objectivity. Thus, the open question is whether the whole system will be reinforced by the bodies that publish the agricultural regulations.

Another essential issue (and highly related to the previous) is the cultural one. Although the technical basis for a semi-automatic compliance checking is mainly available (or under development in the worst case), the tools provided by PA have not yet moved into mainstream agricultural management despite the fact that they are showing promising results. Regarding this issue, I consider that due to increasing suitability to adopt a digital agriculture framework and the reduction of PA costs, huge amount of data will be accessible; and in order to be competitive in the market, farmers must embrace the digitalisation.

Taking into account the results of this thesis and another related works, it is clear that the outlook for machine learning and deep learning in crop protection is very promising. How-

ever, the general weakness in understanding of deep learning in today's marketplace is that of a not being able to formulate holistic and long term solutions to existing problems. DL should not be a hammer where every problem is a nail because this approach will lead to a serious defeat. Deep learning is just a statistical technique, and all statistical techniques suffer from flaws, especially when we do not know their assumptions (e.g.: assumption of normality).

Rather, the ability to craft solutions that integrate deep learning as a component into an integrated solution with another rule-based software components besides human experts will be a more suitable approach. Deep learning excels at solving closed-end classification problems, in which a wide range of inputs must be mapped into a limited number of categories, given that there is enough (high-quality) data available; however, life troubles are not closed-end classification problems, they are not so clearly encapsulated and data appears much more sporadically and with low quality.

Although PA practices and ACC can be used to manage standard farming situations, the farmer is still essential, keeping an eye out for unforeseen situations. In my opinion, humans will always be involved in the whole process but increasingly at a much more strategic level, leaving most operational activities to machines. This point of view can be found in Cybernetics community: to be successful, NLP must blend techniques from a range of fields: linguistics, cognitive science, data science, computer science, domain experts (e.g.: farmers) and more. Only by combination of all possible perspectives, the mystery of the human language phenomena will be bounded; and it could be

integrated within a computer.

Finally, there are some ethical questions that arise from the artificial intelligence-based solutions, as the presented in this thesis. For example, who is responsible for traces of pesticides found on harvested vegetables if the chemical has been applied too late or with an excessive amount? Is it the farmer, any of the software components (e.g., rule extraction system, FMIS, the crop diseases detector), or the manufacturer of any of the used monitoring sensors? Currently there is no clear answer, but we will work to clarify it.

Bibliography

Agacayak, T., Keyman, E. F., 2018. Water and Food Security in Turkey in a Changing Climate. Tech. Rep. March.
URL <http://link.springer.com/10.1007/978-90-481-9974-7>

Amara, J., Bouaziz, B., Algergawy, A., 2017. A Deep Learning-based Approach for Banana Leaf Diseases Classification. In BTW (Workshops), 79–88.

Araujo, D. A. D., Rigo, S. J., Muller, C., Chishman, R., 2013. Automatic information extraction from texts with inference and linguistic knowledge acquisition rules. 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) 3, 151–154.

Athanasiadis, I., Rizzoli, A., Janssen, S., Andersen, E., Villa, F., 2009. Ontology for seamless integration of agricultural data and models. In: Conf. on Metadata and Semantic Research. pp. 282–293.

Baker, T., Caracciolo, C., Arnaud, E., 2016. A Hub for Agricultural Vocabularies. MTSR, 6–8.

Balikas, G., Amini, M.-R., 2016. An empirical study on large scale text classification with skip-gram embeddings Georgios. arXiv.

- Bellman, R., 1961. Adaptive control processes: A guided tour. Princeton University Press 28, 1–19.
URL <http://arxiv.org/abs/1302.6677>
- Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C., 2003. A Neural Probabilistic Language Model. *Journal of Machine Learning Research* 3, 1137–1155.
- Benjamin, C., Gallic, E., 2018. Effects of Climate Change on Agriculture: a European case study.
- Biagioli, C., Francesconi, E., Passerini, A., Montemagni, S., Soria, C., 2005. Automatic semantics extraction in law documents. *Proceedings of the 10th international conference on Artificial intelligence and law - ICAIL '05*, 133.
- Blackmore, S., Godwin, R., Fountas, S., 2003. The Analysis of Spatial and Temporal Trends in Yield Map Data over Six Years. *Biosystems Engineering* (July).
- Blanc, E., Reilly, J., 2017. Approaches to Assessing Climate Change Impacts on Agriculture: An Overview of the Debate. *Review of Environmental Economics and Policy* 11 (2), 247–257.
- Blancard, D., 2012. *Tomato Diseases*.
- Bock, C. H., Poole, G. H., Parker, P. E., Gottwald, T. R., 2010. *Critical Reviews in Plant Sciences Plant Disease Severity Estimated Visually, by Digital Photography and Image Analysis, and by Hyperspectral Imaging*. Taylor & Francis.

- Boella, G., Caro, L. D., Robaldo, L., 2013. Semantic Relation Extraction from Legislative Text using Generalized Syntactic Dependencies and Support Vector Machines. *RuleML-2013*, 218–225.
- Breiman, L., 2001. Random Forests. *Machine learning* 45.1, 5–32.
- Brickley, D., Guha, R., McBride, B., 2014. RDF Schema 1.1. W3C recommendation.
- Brill, E., 1992. Rule-Based Part of Speech. In: *Proceedings of the third conference on Applied natural language*. In: *Proceedings of the third conference on Applied natural language*. pp. 152–155.
- Brillante, L., Gaiotti, F., Lovat, L., Vincenzi, S., Giacosa, S., Torchio, F., Segade, S. R., Rolle, L., Tomasi, D., 2015. Investigating the use of gradient boosting machine, random forest and their ensemble to predict skin flavonoid content from berry physical-mechanical characteristics in wine grapes. *Computers and Electronics in Agriculture* 117, 186–193. URL <http://dx.doi.org/10.1016/j.compag.2015.07.017>
- Carletta, J., 1993. Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics*.
- Caruana, R., Lawrence, S., Giles, L., 2001. Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping. In: *Neural Information Processing Systems Conference*.

- C.E., D., Moss, D., Hill, M., 2004. EUNIS Habitat Classification. Tech. rep., European Environment Agency-European Topic Centre on Nature Protection and Biodiversity.
- Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W. P., 2002. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16, 321–357.
URL <http://www.jair.org/papers/paper953.html>
- Chawla, N. V., Japkowicz, N., Drive, P., 2004. Editorial : Special Issue on Learning from Imbalanced Data Sets Aleksander Kolcz. *ACM SIGKDD Explorations Newsletter* 6 (1), 1–6.
- Chollet, F., 2017. *Deep Learning with Python*, 1st Edition. Manning Publications Co., Greenwich, CT, USA.
- Chomsky, N., 1957. *Syntactic structures*, the hague Edition.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P., 2011. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research* 12, 2493–2537.
- Cortes, C., Vapnik, V., 1995. Support-Vector Networks. *Machine Learning* 20 (3), 273–297.
- Corwin, D., Plant, R., 2005. Applications of apparent soil electrical conductivity in precision agriculture. *Computers and Electronics in Agriculture* 46, 1–10.
- Cowell, L., Smith, B., 2010. Infectious disease ontology. *Infectious Disease Informatics*, 373–395.

- Dalgaard, R., Halberg, N., Kristensen, I. S., Larsen, I., 2006. Modelling representative and coherent Danish farm types based on farm accountancy data for use in environmental assessments. *Agriculture Ecosystems & Environment* 117, 223–237.
- Damos, P., 2013. Semantics and emergent web-technologies: modern challenges for integrated fruit production systems towards internationalization. *IOBC-WPRS Bull* 91, 133–142.
- Damos, P., Karampatakis, S., Bratsas, C., 2017. Representing and integrating agro plant- protection data into semantic web through a crop-pest ontology: the case of the greek ministry of rural development and food (GMRDF) ontology. *IOBC-WPRS Bull.* 123, 122–127.
- Degtyarenko, K., de Matos, P., Ennis, M., 2008. ChEBI: a database and ontology for chemical entities of biological interest. *Nucl. Acids Res.*, 344–350.
- Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv*.
URL <http://arxiv.org/abs/1810.04805>
- DeVries, P., 2013. GeoSpecies Knowledge Base.
- dos Santos, C. N., Xiang, B., Bowen, Z., 2015. Classifying Relations by Ranking with Convolutional Neural Networks. In: 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference

on Natural Language Processing (Volume 1: Long Papers).
No. 1.

Dragoni, M., Villata, S., Rizzi, W., Governatori, G., Dragoni, M., Villata, S., Rizzi, W., Governatori, G., Nlp, C., 2016. Combining NLP Approaches for Rule Extraction from Legal Documents. In: 1st Workshop on Mining and Reasoning with Legal texts (MIREL 2016).

Engers, T. M. V., Gog, R. V., Sayah, K., 2004. A Case Study on Automated Norm Extraction. In: Proc. of JURIX. pp. 49–58.

European Commission, 2005. EU Pesticides Database. Online Database. Tech. rep.

European Commission, 2018. Farm Economy Focus: Spain. Tech. rep.

European Commission, 2018. Statistical Factsheet: Spain. Tech. Rep. May.

Federhen, S., 2012. The NCBI taxonomy database. *Nucleic Acids Research* 40, 136–143.

Ferentinos, K. P., 2018. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture* 145 (September 2017), 311–318.
URL <https://doi.org/10.1016/j.compag.2018.01.009>

Foley, J. A., 2011. Can We Feed the World Sustain the planet? *Scientific American* (November), 60–65.

- Fountas, S., Sorensen, C. G., Tsiropoulos, Z., Gemtos, T., 2014. Farm machinery management information system. *Computers and Electronics in Agriculture*.
- Fountas, S., Wulfsohn, D., Blackmore, B. S., 2006. A model of decision-making and information flows for information-intensive agriculture. *Agricultural Systems* 87, 192–210.
- Francesconi, E., 2010. Legal rules learning based on a semantic model for legislation. In: *Proc. of SPLeT Workshop*. pp. 46–51.
- Friedman, J., Hastie, T., Tibshirani, R., 2008. *The Elements of Statistical Learning. Elements*.
URL <http://www-stat.stanford.edu/~tibs/book/preface.ps>
- Fu, G., Batchelor, C., Dumontier, M., Hastings, J., Willighagen, E., Bolton, E., 2015. PubChemRDF: towards the semantic annotation of PubChem compound and substance databases. *Cheminformatics* 34.
- Garrett, K. A., Dendy, S. P., Frank, E. E., Rouse, M. N., Travers, S. E., 2006. Climate Change Effects on Plant Disease: Genomes to Ecosystems. *Annual Review of Phytopathology* 44 (1), 489–509.
URL <http://www.annualreviews.org/doi/10.1146/annurev.phyto.44.070505.143420>
- Gómez-Pérez, A., Fernández-López, M., Corcho, O., 2004. *Ontological Engineering. Methodologies and Methods for Build-*

- ing Ontologies. In: *Advanced Information and Knowledge Processing*. pp. 125–142.
- Görgens, E. B., Montaghi, A., Rodriguez, L. C. E., 2015. A performance comparison of machine learning methods to estimate the fast-growing forest plantation yield based on laser scanning metrics. *Computers and Electronics in Agriculture* 116, 221–227.
- Goumopoulos, C., Kameas, A., Cassells, A., 2009. An ontology-driven system architecture for precision agriculture applications. *International Journal of Metadata Semantics and Ontologies* 4 (1-2), 72–84.
- Gruber T, 1995. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies* (43), 907–928.
- Habibi, M., Weber, L., Neves, M., Wiegandt, D. L., Leser, U., 2017. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics* 33 (14), i37–i48.
- Hassanpour, S., Connor, M. J. O., DAS, A., 2011. A Framework for the Automatic Extraction of Rules from Online Text. In: *Rule-Based Reasoning, Programming, and Applications - 5th International Symposium, RuleML 2011*. No. July. Barcelona, Spain.
- He, H., Garcia, E. A., 2010. Learning from Imbalanced Data Sets. *IEEE Transactions on knowledge and data engineering* 21 (9), 1263–1264.

- URL <http://www.aaai.org/Papers/Workshops/2000/WS-00-05/WS00-05-003.pdf>
- Hirtle, D., Boley, H., Grosz, B., Kifer, M., Sintek, M., Tabet, S., Wagner, G., 2006. Schema Specification of RuleML. Tech. rep.
- Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural computation* 9 (8), 1–32.
- Horrocks, I., Patel-Schneider, P. F., Boley, H., Said Tabet, M., Grosz, B., Mike Dean, 2007. SWRL: A semantic web rule language combining oWL and ruleML. Tech. rep.
- Huang, Z., Xu, W., Yu, K., 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. CoRR.
URL <http://arxiv.org/abs/1508.01991>
- Information, I. T., Database, S. O.-l., 2010. Integrated Taxonomic Information System.
- J.A., C., 1960. A coefficient of agreement for nominal scales. *Educational And Psychological Measurement*.
- Jain, V. K., Singh, A. K., Bisen, P., Maurya, A. K., Tiwari, A., Pal, S., Varan, R., 2016. Hidden Harvest Under Changing Climate.
URL <https://www.crops.org/publications/cs/abstracts/51/5/2299>
- Jaiswal, P., Avraham, S., Ilic, K., Kellogg, E. A., McCouch, S., Pujar, A., Reiser, L., Rhee, S. Y., Sachs, M. M., Schaeffer,

- M., Stein, L., Stevens, P., Vincent, L., Ware, D., Zapata, F., 2005. Plant Ontology (PO): A controlled vocabulary of plant structures and growth stages. *Comparative and Functional Genomics* 6 (7-8), 388–397.
- Japkowicz, N., Stephen, S., 2002. The class imbalance problem: A systematic study. *Intelligent Data Analysis Journal* 6, 429–450.
- Jones, A., Xu, X., Pittas, N., 2000. Spice: a flexible architecture for integrating autonomous databases to comprise a distributed catalogue of life. In: *Conf. on Database and Expert Systems Applications*. pp. 981–992.
- Jørgensen, L. N., Noe, E., Langvad, A., Rydahl, P., Jensen, J. E., Ørum, J. E., Pinnschmidt, H., Qvist, O., 2007. Vurdering af planteværn onlines økonomiske og miljømæssige effekt (assessment of crop protection online’s economic and environmental effect). *Bekæmpelsesmiddelforskning fra Miljøstyrelsen* 115.
- Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., Mikolov, T., 2017. FastText.zip: compressing text classification models, 1–13.
- Khoo, A., Marom, Y., Albrecht, D., 2006. Experiments with Sentence Classification. *Proceedings of the 2006 Australasian Language Technology Workshop (ALTW2006)*, 18–25.
URL <http://aclweb.org/anthology/U/U06/U06-1005.pdf>

- Kiyavitskaya, N., Zeni, N., Breaux, T. D., Ant, A. I., Cordy, J. R., Mich, L., Mylopoulos, J., 2008. Automating the Extraction of Rights and Obligations for Regulatory Compliance. ER-2008, 1–14.
- Kleinbaum, D. G., Klein, M., 1994. Logistic Regression.
- Kotsiantis, S. B., 2006. Supervised Machine Learning: A Review of Classification Techniques. *Informatica* 31, 249–268.
- Krallinger, M., Leitner, F., Rabal, O., Vazquez, M., Oyarzabal, J., Valencia, A., 2015. CHEMDNER: The drugs and chemical names extraction challenge. *Cheminformatics*.
- Krishnan, V., Ganapathy, V., 2005. Named Entity Recognition.
- Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems*. pp. 1097–1105.
- Labussière, E., Barzman, M., Ricci, P., 2010. European Crop Protection in 2030: A foresight study. Tech. rep.
- Lafferty, J., Mccallum, A., Pereira, F., 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: *Proceedings of ICML*. Vol. 2001. pp. 282–289.
- Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C., 2016. Neural Architectures for Named Entity Recognition. *NAACL-HLT*.

- Langley, P., 1988. Machine learning as an experimental science. *Machine Learning*, 1–5.
URL <https://www.aaai.org/Papers/Workshops/2006/WS-06-06/WS06-06-002.pdf>
- Langley, P., John, G. H., 1995. Estimating continuous distributions in Bayesian classifier. In: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. Montreal, Quebec, pp. 399–406.
- Lecun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. *Nature* 521, 436–444.
- Levenshtein, V. I., 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 707–710.
- Lévy, F., Nazarenko, A., 2013. Formalization of natural language regulations through SBVR structured english. In: *International Workshop on Rules and Rule Markup*. pp. 19–33.
- Li, W. W., Byrnes, R. W., Bourne, J. H., Birnbaum, A., Reyes, V. M., Shahab, A., Mosley, C., Pekurovsky, D., Shindyalov, G. B. Q., N., I., Casanova, H., Ang, L., Berman, F., Arzberger, P. W., Miller, M. A., Bourne, P. E., 2004. The encyclopedia of life project: grid software and deployment. *New Gener. Comput.* 22 (2), 127–136.
- Ling, W., Lu, T., Ram, M., Amir, S., Dyer, C., Black, A. W., Trancoso, I., 2015. Finding Function in Form: Compositional

- Character Models for Open Vocabulary Word Representation (September), 1520–1530.
- Lipton, Z. C., Berkowitz, J., Elkan, C., 2015. A Critical Review of Recurrent Neural Networks for Sequence Learning arXiv : 1506 . 00019v4 [cs . LG] 17 Oct 2015. CoRR, 1–38.
- Liu, L., Shang, J., Ren, X., Xu, F. F., Gui, H., Peng, J., Han, J., 2017. Empower Sequence Labeling with Task-Aware Neural Language Model. CoRR.
- Luo, G., Huang, X., Lin, C., Nie, Z., 2015. Joint Named Entity Recognition and Disambiguation. In: Proceedings of EMNLP. No. September. pp. 879–888.
- Luo, L., Yang, Z., Yang, P., Zhang, Y., Wang, L., Lin, H., Wang, J., 2018. An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition. *Bioinformatics* 34 (8), 1381–1388.
- Ma, X., Hovy, E., 2016. End-to-end Sequence Labeling via Bidirectional LSTM-CNNs-CRF. CoRR.
URL <http://arxiv.org/abs/1603.01354>
- Maat, E. D., Winkels, R., 2010. Suggesting Model Fragments for Sentences in Dutch Law. In: LOAIT. pp. 19–28.
- Malarkodi, C. S., Lex, E., Devi, S. L., 2016. Named Entity Recognition for the Agricultural Domain. *Research in Computing Science* 117, 121–132.

- McCallum, A., Freitag, D., Pereira, F., 2000. Maximum Entropy Markov Models for Information Extraction and Segmentation. International Conference on Machine Learning (ICML), 591–598.
- McGuinness, D., Van Harmelen, F., 2004. OWL Web Ontology Language Overview. W3C Recommendation.
- Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient Estimation of Word Representation in Vector Space Topics. In: Proceedings of the Workshop at ICLR.
- Mikolov, T., Kombrink, S., Burget, L., Cernocky, J., Khudanpur, S., 2011. Extensions of Recurrent Neural Network Language Model. In: IEEE (Ed.), Acoustics, Speech and Signal Processing (ICASSP).
- Mohanty, S. P., Hughes, D. P., Salathé, M., 2016. Using Deep Learning for Image-Based Plant Disease Detection. *Frontiers in Plant Science* 7 (September), 1–10.
URL <http://journal.frontiersin.org/Article/10.3389/fpls.2016.01419/abstract>
- Nash, E., Wiebensohn, J., Nikkilä, R., Vatsanidou, A., Fountas, S., Bill, R., 2011. Towards automated compliance checking based on a formal representation of agricultural production standards. *Computers and Electronics in Agriculture* 78 (1), 28–37.
URL <http://dx.doi.org/10.1016/j.compag.2011.05.009>
- Natural Resource Conservation Service, 2016. The Plants Database.

- Nazir, N., Bilal, S., Bhat, K., Shah, T., Badri, Z., Bhat, F., Wani, T., Mugal, M., Parveen, S., Dorjey, S., 2018. Effect of Climate Change on Plant Diseases. *International Journal of Current Microbiology and Applied Sciences* 7 (06), 250–256.
- Ng, A., Coates, A., Saxe, A., Maas, A., Manning, C., Ngiam, J., Socher, R., Le, K., 2013. Machine Learning and AI via Brain simulations. Tech. rep.
- Nguyen, T. H., 2017. Deep learning for information extraction. Ph.D. thesis.
- Nikkilä, R., Wiebensohn, J., Nash, E., Seilonen, I., Koskinen, K., 2012. A service infrastructure for the representation, discovery, distribution and evaluation of agricultural production standards for automated compliance control. *Computers and Electronics in Agriculture* 80, 80–88.
- Oerke, E., 2006. Crop losses to pests. *Journal of Agricultural Science*, 31–43.
- Palkar, R. K., Gala, K. D., Shah, M. M., Shah, J. N., 2016. Comparative Evaluation of Supervised Learning Algorithms for Sentiment Analysis of Movie Reviews. *International Journal of Computer Applications* 142 (1), 975–8887.
URL <http://www.ijcaonline.org/archives/volume142/number1/palkar-2016-ijca-909660.pdf>
- Park, E., Jang, S., Choi, S., Kim, J., 2018. Text Classification using Capsules. *CoNLL* (August).

- Passos, A., Kumar, V., Mccallum, A., 2014. Lexicon Infused Phrase Embeddings for Named Entity Resolution. In: Proceedings of CoNLL. pp. 78–86.
- Patil, S., Pawar, S., Palshikar, G., 2013. Named Entity Extraction using Information Distance. Proceedings of the Sixth International Joint Conference on Natural Language Processing (October), 1264–1270.
URL <http://aclweb.org/anthology/I13-1180>
- 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., Duchesnay, E., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, 2825–2830.
- Peters, M. E., Gardner, M., Neumann, M., Iyyer, M., Clark, C., Lee, K., Zettlemoyer, L., 2018. Deep contextualized word representations. arXiv.
- Plant Ontology Consortium, 2002. The Plant Ontology Consortium and Plant Ontologies 3 (2), 137–142.
- Porter, M., 1980. An algorithm for suffix stripping. *Program*, 130–137.
- Pustejovsky, J., Stubbs, A., 2013. *Natural Language Annotation for Machine Learning*. O’reilly.

- Qi, P., Luo, X., Zhang, D., 2009. Weed recognition based on digital image processing in wheat field. *Natural Science Education*, 136–137.
- Raschka, S., 2014. Naive Bayes and Text Classification I - Introduction and Theory, 20.
URL <http://arxiv.org/abs/1410.5329>
- Ratinov, L., Roth, D., 2009. Design Challenges and Misconceptions in Named Entity Recognition. In: *Proceedings of CoNLL*. No. June. pp. 147–155.
- Rehman, A., Shaikh, Z., 2011. Ontagri: scalable service oriented agriculture ontology for precision farming. In: *Conf. on Agricultural and Biosystems Engineering*. pp. 1–2.
- Rodriguez-Iglesias, A., Rodriguez-Gonzalez, A., Irvine, A., Sesma, A., Urban, M., Hammond-Kosack, K. E., Wilkinson, M. D., 2016. Publishing fair data: an exemplar methodology utilizing Phi-Base. *Frontiers in Plant Science*.
- Rong, X., 2016. word2vec Parameter Learning Explained.
- Ruder, S., 2017. An overview of gradient descent optimization. *CoRR*.
- Saad, O., Darwish, A., Faraj, R., 2012. A survey of machine learning techniques for Spam filtering 12 (2), 66–73.
- Sabour, S., Frosst, N., Hinton, G. E., 2017. Dynamic Routing Between Capsules. In: *Neural Information Processing Systems*. No. Nips.
URL <http://arxiv.org/abs/1710.09829>

- Schuster, M., Paliwal, K. K., 1997. Bidirectional Recurrent Neural Networks. *IEEE Transactions on signal processing* 45 (11), 2673–2681.
- Schut, M., Rodenburg, J., Klerkx, L., Ast, A. V., Bastiaans, L., 2014. Systems approaches to innovation in crop protection . A systematic literature review. *Crop Protection* 56, 98–108. URL <http://dx.doi.org/10.1016/j.cropro.2013.11.017>
- Segura-Bedmar, I., Martinez, P., Herrero-Zazo, M., 2013. Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013). In: *Proceedings of the 7th International Workshop on Semantic Evaluation*. Vol. 2. pp. 341–350.
- Sharnagat, R., 2014. Named Entity Recognition: A Literature Survey. Tech. rep. URL <http://www.cfilt.iitb.ac.in/resources/surveys/rahul-ner-survey.pdf>
- Shi, L., Roman, D., 2017. From standards and regulations to executable rules : A case study in the Building Accessibility domain. *CEUR Workshop Proceedings*, 1–10.
- Siddiqui, M. A., 2016. An empirical evaluation of text classification and feature selection methods. *Artificial Intelligence Research* 5 (2).
- Sigrimis, N., Arvanitw, K. G., Pasgianos, G., Anastasiou, A., Ferentinos, K. P., 2000. New ways on supervisory control: a virtual greenhouse: to train, to control and to manage. In:

- AC Symposium on Manufacturing Modelling, Management and Control (MIM 2000). Rio, Greece, pp. 561–568.
- Sini, M., 2009. Semantic technologies at FAO, agricultural information management standards. In: International Society for Knowledge Organization: ISKO.
- Skalidis, S., 2016. Adaptive Synthetic Sampling Approach for Imbalanced Learning (3), 1322–1328.
- Skofic, M., Arnaud, E., McLaren, G., Matteis, L., Portugal, A., Hyman, G., Shrestha, R., 2012. Bridging the phenotypic and genetic data useful for integrated breeding through a data annotation using the Crop Ontology developed by the crop communities of practice. *Frontiers in Physiology* 3 (August), 1–10.
- Sørensen, C. G., Pesonen, L., Fountas, S., Suomi, P., Bochtis, D., Bildsøe, P., Pedersen, S. M., 2010. A user-centric approach for information modelling in arable farming. *Computers and Electronics in Agriculture* 73 (1), 44–55.
- Soria, C., Bartolini, R., Lenci, A., Pirrelli, V., 2005. Automatic extraction of semantics in law documents. European Press Academic Publishing.
- Stellato, A., 2002. The AGROVOC Linked Dataset. *Semantic Web Journal*.
- Strubell, E., 2017. Dependency Parsing with Dilated Iterated Graph CNNs. In: Conference on Empirical Methods in Natural Language Processing.

- Tang, J. L., Wang, D., Zhang, Z. G., He, L. J., Xin, J., Xu, Y., 2017. Weed identification based on K-means feature learning combined with convolutional neural network. *Computers and Electronics in Agriculture* 135, 63–70.
URL <http://dx.doi.org/10.1016/j.compag.2017.01.001>
- Tomek, I., 1976. Two Modification of CNN. *IEEE Transactions on Systems, Man, and Cybernetics* 6 (11), 769–772.
- Toutanova, K., Klein, D., Manning, C. D., Singer, Y., 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - NAACL '03* 1, 173–180.
URL <http://portal.acm.org/citation.cfm?doid=1073445.1073478>
- Ur Rahman, H., Hahn, T., Segall, R., 2016. Disease named entity recognition using conditional random fields. *CEUR Workshop Proceedings* 1650, 37–41.
- W3C, 2009. RIF Overview. Tech. rep., W3C.
- Walls, R., Athreya, B., Cooper, L., 2012a. Ontologies as integrative tools for plant science. *American Journal of Botany* 99 (8), 263–1275.
- Walls, R., Smith, B., Elser, J., Goldfain, A., Stevenson, D. W., Jaiswal, P., 2012b. A plant disease extension of the infectious disease ontology. *ICBO*, 1–5.

- Wang, Y., Sun, A., Han, J., 2018. Sentiment Analysis by Capsules. In: WWW. Vol. 2. pp. 1165–1174.
- Wegener, J. K., 2017. New technical solutions for precise and safe application of plant protection products, 417–421.
- Wei, Q., Chen, T., Xu, R., He, Y., Gui, L., 2016. Disease named entity recognition by combining conditional random fields and bidirectional recurrent neural networks. Database 2016 (December), 1–8.
- Welch, B. L., 1951. On the Comparison of Several Mean Values: An Alternative Approach.
URL <http://www.jstor.org/stable/2332579>
- Wikimedia Foundation, 2017. Wikispecies: Free Species Dictionary.
- Wolpert, D. H., Macready, W. G., 1995. No Free Lunch Theorems for Optimization. Tech. rep.
- Wu, L., Liu, J., Wen, Y., Deng, X., 2009. Weed identification method based on SVM in the corn field. Trans. Chin. Soc. Agric. Mach 40 (1), 162–166.
- Wyner, A., Peters, W., Science, C., 2011. On Rule Extraction from Regulations. In: JURIX-2011. pp. 113–122.
- Yang, J., Liang, S., Zhang, Y., 2018. Design Challenges and Misconceptions in Neural Sequence Labeling. CoRR.

- Yano, K., 2018. Neural Disease Named Entity Extraction with Character-based BiLSTM+CRF in Japanese Medical Text. arXiv (2003).
URL <http://arxiv.org/abs/1806.03648>
- Zhang, H., 2004. The Optimality of Naive Bayes. Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference FLAIRS 2004 1 (2), 1 – 6.
URL <http://www.aaai.org/Papers/FLAIRS/2004/Flairs04-097.pdf>
- Zhang, J., 2015. Automated Extraction of Information from Building Information Models into a Semantic Logic-Based Representation. In: International Workshop on Computing in Civil Engineering, No. November.
- Zhang, Q., Pierce, F. J., 2013. Agricultural Automation: Fundamentals and Practices. CRC Press.
- Zhou, R., Kaneko, S., Tanaka, F., Kayamori, M., Shimizu, M., 2014. Disease detection of Cercospora Leaf Spot in sugar beet by robust template matching. Computers and Electronics in Agriculture 108, 58–70.
URL <http://dx.doi.org/10.1016/j.compag.2015.05.020>