End-to-End Sequence Labeling via Deep Learning for automatic extraction of agricultural regulations

Borja Espejo-Garcia[a], Francisco J. Lopez-Pellicer[a], Javier Lacasta[a], Ramón Piedrafita Moreno[a] and F. Javier Zarazaga-Soria[a]

[a]Computer Science and System Engineering Department, Universidad Zaragoza, María de Luna 1, 50018, Zaragoza, Spain

Emails: {borjaeg, fjlopez, lacasta, piedrafi, javy}@unizar.es

Corresponding author: Borja Espejo-Garcia. Email: borjaeg@unizar.es

Abstract

In the European Union, production standards in the form of legal regulations play an important role in farming. Because of the increasing amount of regulations, it is desirable to transform human-oriented regulations into a set of computer-oriented rules to provide decision support through the Farm Management Information System. To obtain the logical structure of rules, automatically labeling their meaningful information is necessary.

In this work, we evaluate the performance of 8 different state-of-the-art deep learning architectures to develop an end-to-end sequence labeler for phytosanitary regulations. This sequence labeler extracts different meaningful information items to determine which pesticides can be applied to a crop, the place of the treatment, when it can be applied, and the maximum number of applications. The architectures evaluated do not require feature engineering and, hence, they are applicable to the agricultural regulations of different countries. The best system is a neural network that uses character embeddings, Bidirectional Long short-term memory and Softmax. It achieves a performance of 88.3% F_1 score.

Keywords

- Deep Learning
- Natural Language Processing
- Rule extraction
- Compliance Control
- Sequence Labeling

1. Introduction

Agricultural production is governed by regulations, such as phytosanitary authorizations and Integrated Pest Management guidelines, that restrict farming practices harmful for the environment (Nikkilä et al., 2012). For instance, the phytosanitary authorizations contain obligations and prohibitions about the place where a treatment can be applied (greenhouse or outdoors) and the amount of pesticide that can be supplied. To avoid information overload, farm software, such as a Farm Management Information System (FMIS), should have a module that guarantee compliance control in operations managed (e.g. spraying pesticides) by ensuring that such operations are conform to the relevant regulations. This module would be easier to implement if regulations were available in both machine-readable and natural language formats (Nash et al., 2011). FMIS refer to a planned system for collecting, processing, storing and disseminating data in the form needed to carry out a farm's operations and functions (Salami and Ahmadi, 2010). Currently, their functionalities are useful in farms that seek to comply with agricultural standards while maintaining high product quality. However, to meet reliably all these needs, there is a double bottleneck: firstly, relevant information is scattered among different sources and, secondly, it is only available through texts in natural language. For this reason, making them computer-oriented require an analysis of their objectivity, interpretability and feasibility to transform it into a formal representation as a statement in first-order logic (Nash et al., 2011). In order to obtain the logical structure of the regulations, it is necessary to label previously all the relevant information used within the definition of the rule (e.g. crops, phenological stages, temporal relations). However, because of the increasing amount of regulations, automatic techniques supporting the labeling task are desirable.

Although legal language is considerably more constrained than ordinary language, its syntactic structures require labeling approaches using techniques related to Natural Language Processing (NLP) and machine learning (ML) (Maat and Winkels, 2008; Brighi and Palmirani, 2009; Wyner and Peters, 2011; Dragoni et al., 2016). However, despite the huge number of proposed approaches, the problem of extracting rules is still open. Most traditional high performance sequence labeling models are linear statistical models, including Hidden Markov Models and Conditional Random Fields (CRF) (Ratinov and Roth, 2009; Passos et al., 2014; Luo et al., 2015), which rely heavily on hand-crafted features and task-specific resources such as orthographic features and dictionaries. However, such task-specific knowledge is costly to develop (Ma and Xia, 2014), making sequence labeling models difficult to adapt to new tasks, domains or languages.

With advances in deep learning, neural network models have reached state-of-the-art results on many sequence labeling tasks (Lample et al., 2016; Ma and Hovy, 2016). Different neural architectures such as Convolutional Neural Network (CNN), recurrent neural networks (Goller and Kuchle, 1996), together with its variants such as Long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) have gained prominence in past few years. Collobert et al., 2011 proposed a seminal neural architecture for sequence labeling. It captures word sequence information with a one-layer CNN based on pre-trained word embeddings, followed with a CRF output layer (Lafferty and Mccallum, 2001). Dos Santos et al., 2015 extended this model by integrating character-level CNN features. LSTMs have shown great success solving sequence labeling tasks and achieving competitive performance against traditional models (Graves et al., 2013; Huang et al., 2015; Hu et al., 2016).

Deep learning has recently entered in the domain of agriculture. Different areas have introduced this method to improve their efficacy. Some of them are highly related to plant protection; for example, general recognition of plant diseases by leaf image classification with CNNs (Sladojevic et al., 2016; Mohanty et al., 2016) and transfer learning (Ferentinos, 2018) or specific ones such as disease recognition in banana leaves (Amara et al., 2017). There are also works in weed identification (Xinshao and Cheng, 2016; Tang et al., 2017). More works can be found in Kamilaris and Prenafeta-Boldú, 2018.

However, the state of the art represented by the previous works, focuses on computer vision techniques. Nowadays, there are very few works in literature that focus on information extraction and the automatic text labeling from regulations. For example, Patil et al., 2013 used unsupervised learning to extract crops, diseases and chemical treatments. Malarkodi et al., 2016 proposed an approach for labeling crops, chemicals, locations and temperatures among others by using CRF. None of these examples have used deep learning techniques.

In this work, our goal is to find a suitable deep-learning architecture for building an end-to-end information extraction system in agricultural regulations. For this reason, we evaluate 8 state-of-the-art neural network architectures to label the meaningful parts of textual rules found in a phytosanitary products registry (phenological stages, maximum number of applications, temporal relations, etc.). The phytosanitary products regulations are especially relevant for society because sustainable plant protection is a pressing challenge (**?**). Moreover, in the Spanish phytosanitary authorizations, regulations can be structured into a set of individual rules which roughly fulfill 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. We do not have 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 precision agriculture such as remote sensing, computer vision and FMIS.

Our contribution is to show an end-to-end methodology to automatically label meaningful parts of the phytosanitary regulations using a deep leaning model. A model that does not require task-specific resources, feature engineering, or data preprocessing beyond pre-trained word embeddings on unlabeled corpora. Thus, our approach can be applied to a wide range of sequence labeling tasks on different agricultural regulations of different countries. We explore three neural model design decisions: character sequence representations, word sequence representations and inference layer. We combine character and word-level representations and feed them into bi-directional LSTM (BLSTM) and CNN to model context information of each word. On top of BLSTM and 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. This research is the continuation of (**?**), where the categorization of rules as obligations or prohibitions was evaluated.

In this work, we go one step further and we evaluate different deep learning architectures in order automatically label the meaningful parts of the rules.

This article is structured as follows: Section 2 presents the different deep neural networks that we are going to evaluate. Section 3 explains the experimental setup. Section 4 shows the results and Section 5 presents a discussion. Finally, in Section 6 conclusions are drawn and future directions envisaged.

2. Material and Methods

2.1. Neural Network Architectures

In this section, we describe the layers of the neural network architectures evaluated in this research. 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 1.

Character Sequence Representations. Character features such as prefix, suffix and capitalization 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 critical for successful sequence labeling tasks. In this work, we study the impact of using character embeddings randomly initialized. The character sequence information is captured with LSTM.

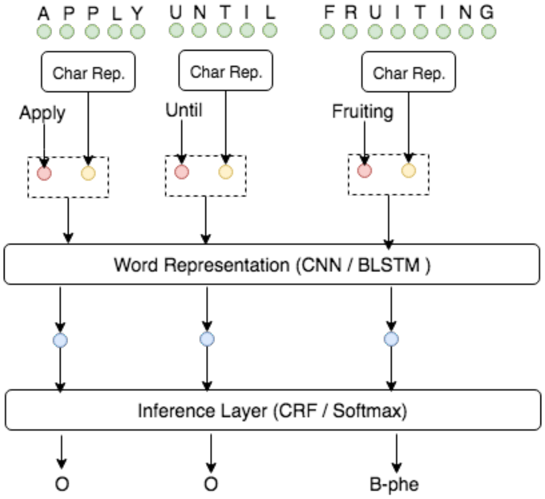


Figure 1: Neural sequence labeling architecture for sentence "Apply until flowering". Green, red, yellow and blue circles represent character embeddings, word embeddings, character sequence representations and word sequence representations, respectively.

Word Sequence Representations. In order to represent individual words, supervised labeling models require large amounts of manually labeled data to achieve good performance. This data is hard to acquire. However, it is possible to take advantage of very large unlabeled data to learn word features to enrich simpler models obtained from small gold standards. These representations are called word embeddings (Mikolov et al., 2011) and they perform well across a variety of tasks (Collobert et al., 2011). Word embedding is a type of mapping that allows words with similar meaning to have similar representation. In this work, we use pre-trained Word2vec with Skipgram (Rong, 2016) to initialize word embeddings. Word sequence representations are concatenated with character sequence representations.

Similar to character sequences, we can model word sequence information through LSTM or CNN structures. LSTMs are a family of neural networks that operate on sequential data that effectively capture long-range dependencies (Liu et al., 2017; Lample et al., 2016; Ma and Hovy, 2016). Unidirectional LSTMs suffer from weakness of not utilizing the future contextual information. However, bidirectional LSTM (BLSTM) addresses this 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.

On the other hand, CNNs combine different 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 takes the extracted word sequence representations as features and assigns labels to the word sequence. A very simple but effective labeling model is to use the hidden layer as features to make independent labeling decisions for each output by using a Softmax layer (Ling et al., 2015). 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 natural language processing 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, a “grammar” that captures the correlations between adjacent labels imposes constraints impossible to model with Softmax even when BLSTM is used. For this reason, statistical linear models such as CRF can be used. We compare the use of Softmax and CRF as inference layer implementation.

3. Experiments

This section describes the experiments performed on end-to-end neural networks. Although the process presented in this work, is suitable for labeling 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.

3.1. Spanish Case

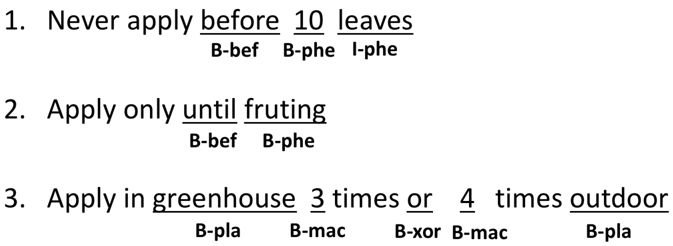
In order to create a sequence labeler with deep learning, the labeler must be previously trained with a gold corpus, which is a set of manually annotated texts that serves as a basis for the training and evaluation. In this work, we use a corpus focused on the official information about crops and authorized pesticides from the Spanish phytosanitary products registry. Currently, this repository stores 2,426 PDFs that contain authorizations and instructions about how to apply phytosanitary products to comply with environmental regulations. Moreover, 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 paper.

To address the annotation task, we have followed the MATTER methodology (**?**). With this methodology, the first step is to model the phenomena that is going to be extracted from texts through the label types shown in Table 1. In our case, we model the application of phytosanitary product, where the maximum number of applications, phenological stages or place where the treatment can be applied must be explicitly specified based on official documents (2,426 PDFs). Moreover, there are two label types ("xor" and "rep") that do not appear in the Table 1, but they are necessary to label information accurately; "xor" is used when the treatment has two ways of being applied, and "rep" is used when the treatment has several 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 was studied in our previous work. Since some concepts are expressed through multi-word expressions (continuous sequences of words), the "IOB" 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, that if the word do not refer to an event). The corpus statistics are shown in Table 2 and Table 3. The use of a large corpus is especially important 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 corpus is publicly available [1].

[1] https://github.com/borjaeg/ner/blob/master/ner_corpus_condicionamientos_revisited_filtered.csv

Table 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



1. Never apply before 10 leaves
2. Apply only until fruting
3. Apply in greenhouse 3 times or 4 times outdoor

Figure 2: Example of labeled rules

Table 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 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

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.

3.2. Hyperparameters

Hyperparameters including learning rate, hidden layer size and number of layers can strongly affect model performance. Table 4 summarizes the chosen hyperparameters for our experiments. These hyperparameters for our models were tuned on the development set by random search. 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). Parameter optimization is performed with stochastic gradient descent (SGD) with batch size 1 and a fixed learning rate. We explore other more sophisticated optimization algorithms such as RMSProp and AdaDelta (Ruder, 2017), but in preliminary experiments they did not improve upon plain SGD. We use early stopping (Caruana et al., 2001) based on performance on validation sets. Pre-trained word embeddings are evaluated with fine-tuning. For the implementation of the neural networks, we use Keras 2.1.6 (Chollet, 2017).

Table 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
Learning Rate	0.1

3.3. Evaluation

F_1 score (Eq. 1) is used as the evaluation metric for sequence labeling, where precision is the ratio of correct labels in the sequence labeler 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 labeler 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 3 for more details). To reduce the volatility of the system, we conduct each experiment 5 times under different random seeds, and report the mean for each neural architecture.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

We also present a comparison of the performance of each of the main components of the neural architectures. This evaluation can clarify which are the more promising future research directions.

4. Experimental 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.

Table 5: Nomenclature used to describe neural architectures in table 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

4.1. Evaluation per label type

Examining the results in Table 6, we think that there are some important facts to remark. Each label has its own best sequence labeler (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. 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 3, these labels contain few examples and deep learning 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 labeling more tricky. Finally, in the last table row, we show the micro-average F_1score for each neural network architecture. Taking into account this result, the C-B-S architecture shows the highest performance and it can be considered as the best overall approach. 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 note that all the neural networks evaluated obtain performances over 82%, which are results quite similar to those obtained in the community sequence labeling benchmarks (Malarkodi et al., 2016; Patil et al., 2013).

Table 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

4.2. Network Settings

Following the framework proposed by Yang et al., 2018, the aim of this part of the evaluation is the comparison of the different neural layers to study which ones lead to an overall better performance. In order to confirm that the differences are not due to chance, we have computed statistical significance test. Specifically, the test has been performed using the Welch's t-test (Welch, 1951). We have set the the confidence level to 0.1.

Char vs No char. In our experiments, according to Figure 3 a), character information slightly improves the sequence labeling models. Moreover, the difference is statistically significant ($p < 0.1$).

CNN vs BLSTM. In Figure 3 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 labeling.

CRF vs Softmax. According to Figure 3 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 is our corpus. Moreover, the difference is not statistically significant ($p > 0.1$).

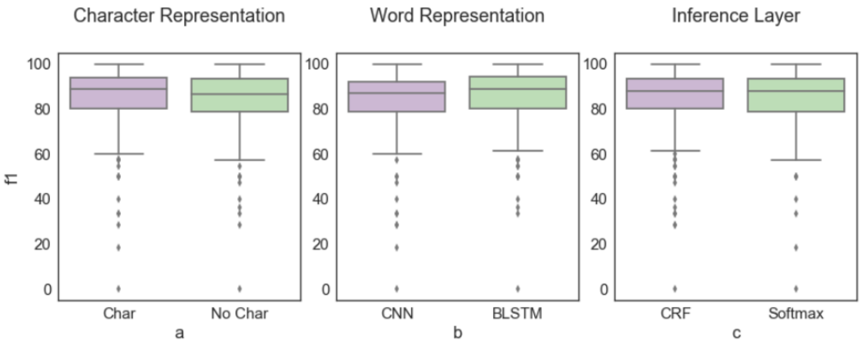


Figure 3: Comparison of layers performance

5. Discussion

In this work, we have developed an end-to-end sequence labeler, which is a necessary step to automate the translation of human-oriented regulations into formal rules. Moreover, because deep neural networks automatically extract language-independent features, this approach can be applied to transform regulations in other countries. In our experiments, we have found that an architecture with character embeddings, BLSTM and Softmax obtains the best performance. This system, with an overall performance of 88.3% F_1score, overcomes the rest of the approaches. Currently, there is no benchmark for sequence labeling in the context of agricultural regulations and, therefore, it is difficult to directly compare our results with another works. However, there are two related works that are important to mention. 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, we can conclude that our results are quite similar to the state-of-the art works in the agricultural domain.

It is important to note that all the architectures presented in this paper 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 achieve better or worse performance, but we can conclude that none of the layers always improves decisively the performance of the sequence labeler. 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 hypothesize 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.

6. Conclusions

In this paper, we have presented an empirical evaluation of 8 state-of-the-art deep learning architectures to develop and end-to-end sequence labeler for the phytosanitary regulations in agriculture domain. To the best of our knowledge, this work is the first attempt in sequence labeling of phytosanitary regulations by using a systematic comparison of different deep learning 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. In our experiments, the best sequence labeler system has a character embedding layer as the character sequence representation, a BLSTM as the word sequence representation and a Softmax as the inference layer. This architecture achieves 88.3% F_1score, which is similar to results obtained in related work.

Despite the good results, we believe that the performance can be further improved. Deep learning 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 convolutional neural network for labeling will be evaluated. This architecture involves using multiple versions of the standard model with different sized kernels. This allows the document to be modeled 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.

7. Acknowledgement

The work of Borja Espejo-Garcia has been partially supported by the Government of Aragon and the European Social Fund through the grant number [C38/2015]. This work has also been supported by the Spanish Government (projects RTC-2016-4790-2 and TIN2017-88002-R). We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Quadro P6000 GPU used for this research.

8. References

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

Brighi, R., Palmirani, M., 2009. Legal Text Analysis of the Modification Provisions: a Pattern Oriented Approach. In: Proceedings of the 12th International Conference on Artificial Intelligence and Law. pp. 238–239.

Caruana, R., Lawrence, S., Giles, L., 2001. Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping. In: Neural Information Processing Systems Conference.

Chollet, F., 2017. Deep Learning with Python, 1st Edition. Manning Publications Co., Greenwich, CT, USA.

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.

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., 2016. Combining NLP Approaches for Rule Extraction from Legal Documents. In: 1st Workshop on MIning and REasoning with Legal texts (MIREL).

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

Goller, C., Kuchle, A., 1996. Learning task-dependent distributed representations by backpropagation through structure. In: Neural Networks IEEE International Conference.

Graves, A., Mohamed, A., Hinton, G., 2013. Speech Recognition With Deep Recurrent Neural Networks. In: ICASSP. No. 3.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural computation 9 (8), 1–32.

Hu, Z., Ma, X., Liu, Z., Hovy, E., Xing, E. P., 2016. Harnessing Deep Neural Networks with Logic Rules. CoRR, 2410–2420.

Huang, Z., Xu, W., Yu, K., 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. CoRR.
URL http://arxiv.org/abs/1508.01991

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., Mikolov, T., 2017. FastText.zip: compressing text classification models, 1–13.

Kamilaris, A., Prenafeta-Boldú, F. X., 2018. Deep learning in agriculture: A survey. Computers and Electronics in Agriculture 147 (July 2017), 70–90.

Lafferty, J., Mccallum, A., 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data 2001 (June), 282–289.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C., 2016. Neural Architectures for Named Entity Recognition. NAACL-HLT.

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.

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.

Ma, X., Hovy, E., 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. CoRR.
URL `http://arxiv.org/abs/1603.01354`

Ma, X., Xia, F., 2014. Unsupervised Dependency Parsing with Transferring Distribution via Parallel Guidance and Entropy Regularization. In: ACL. pp. 1337–1348.

Maat, E. D., Winkels, R., 2008. Automatic Classification of Sentences in Dutch Laws. In: Proceedings of the 2008 conference on Legal Knowledge and Information Systems: JURIX 2008: The Twenty-First Annual Conference. Vol. 6036. Berlin, pp. 207–216.

Malarkodi, C. S., Lex, E., Devi, S. L., 2016. Named Entity Recognition for the Agricultural Domain. Research in Computing Science 117, 121–132.

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`

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.

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

Ratinov, L., Roth, D., 2009. Design Challenges and Misconceptions in Named Entity Recognition. In: Proceedings of CoNLL. No. June. pp. 147–155.

Rong, X., 2016. word2vec Parameter Learning Explained.

Ruder, S., 2017. An overview of gradient descent optimization. CoRR.

Salami, P., Ahmadi, H., 2010. Review of Farm Management Information Systems (FMIS). New York Science Journal Review of FMIS New York Science Journal 3 (5), 87–95.

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., Stefanovic, D., 2016. Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. Computational Intelligence and Neuroscience 2016.

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

Welch, B. L., 1951. On the Comparison of Several Mean Values: An Alternative Approach.
URL http://www.jstor.org/stable/2332579

Wyner, A., Peters, W., 2011. On rule extraction from regulations. Frontiers in Artificial Intelligence and Applications 235, 113–122.

Xinshao, W., Cheng, C., 2016. Weed seeds classification based on PCANet deep learning baseline. 2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2015 (December), 408–415.

Yang, J., Liang, S., Zhang, Y., 2018. Design Challenges and Misconceptions in Neural Sequence Labeling. CoRR.