Applied Soft Computing 95 (2020):106496

DOI: https://doi.org/10.1016/j.asoc.2020.106496

Journalistic transparency using CRFs to identify the reporter of newspaper articles in Spanish

Francisco Jurado[a,*]

[a] *Universidad Autónoma de Madrid, Francisco Tomás y Valiente 11, 28049 Madrid (Spain)*

Abstract

Journalistic transparency rises as a key issue against the lack of credibility to which journalists are exposed, as well as the media manipulators and fake news providers. With the use of Natural Language Processing (NLP) and Machine Learning (ML), it is possible to automate the extraction of information from newspaper articles to know what the sources of information are to verify their veracity. Along with this article, we present the application of Conditional Random Fields (CRFs) for a specific type of Entity Recognition (ER) task, namely, to identify what we have called the *"reporter"* in newspaper articles, i.e., who or what is the provider of the information. Thus, we have created a labelled corpus for the Spanish language and trained and analyzed several CRFs models with a set of specific features. The obtained results suppose a solid baseline for our goal.

Keywords: Journalistic Transparency, Conditional Random Fields, Entity Extraction

1. Introduction

Nowadays, the lack of credibility, the manipulative media and the problem of fake news [1] make the news media vulnerable to scrutiny, and journalistic transparency emerges as a key issue [2, 3, 4, 5]. In this situation, to corroborate information by directly going to the source results essential.

Hence, this article aims to use Natural Language Processing (NLP) and Machine Learning (ML) techniques to make possible the automatic extraction of relevant information from newspaper articles to know what the sources of information are to verify their veracity.

This task does not involve only the Named Entity Recognition (NER) to extract the designators in the text such as proper nouns and temporal expressions [6], but it implies the use of Entity Recognition (ER). In contrast to NER, where the name of entities (organization, person, location, etc.) is detected, the ER task aims to detect the entities in documents to improve the performance of some high-level NLP tasks like Question Answering, Auto Summarization, Machine Translation, and Information Retrieval [7, 8].

*Corresponding author
Email address: `Francisco.Jurado@uam.es` (Francisco Jurado)

Accordingly, to allow the verification of the information that we can find in the written text by spotting the source of that information, the high-level NLP question we want to answer is *"who says that?"*. Thus, we can identify the source of information provided in the newspaper, that is, label what we have called the *"reporter"* in newspaper articles. More particularly, we will perform this task for the Spanish language.

To clarify, we need to pay special attention to the word *"reporter"* due to this word has several meanings in English. Among these meanings, the most commonly used is the journalist who gathers information, investigates, and writes news for different media. However, in this case, the first meaning of the online dictionary Wiktionary[1] will be used, which defines a reporter as *"someone or something that reports"*, i.e., we will refer to *"reporter"* as the person, company, media, report, bulleting, etc. that reports the information.

To face this issue, we must take into account that journalists use to write their news providing a lot of information in one sentence, and also they use to do it using many different (and sometimes really complex) grammatical structures. This makes not easy the process to identify the reporter.

To better exemplify the issue, Table 1 shows some sentences in English with their corresponding translation in Spanish. Particularly:

Example 1. This example shows the easiest way to find the reporter. The sentence is written in the direct speech (or quoted speech), where we can identify who exactly provides the information.

Example 2. In this example, we can see another easy way to find the reporter. In this case, it is the subject of a indirect speech (or reported speech) sentence.

Example 3. This is an example where the sentence in the reported speech starts after the comma, and the real reporter appears just before it. Therefore, we have to ignore *Court* due to the real reporter is *High Court of Justice.*

Example 4. This time, the role of the reporter appears together with a named entity (the name of the organization), but the entity we are interested in (the right reported) is the one surrounded by commas.

Example 5. In this case, the reporter is the *Official Gazette* in charge of publishing the information, but not a specific person or organization.

Example 6. This example shows another typical situation where more than one reporter appears, in this particular two geopolitical locations (countries).

Example 7. This sentence shows a situation where a lot of named entities appear, but only some of them are the appropriated. Firstly, we find the report that includes the information, later the organization that provides the report, and finally, the group that informs the media.

[1] https://en.wiktionary.org/wiki/reporter#English

Table 1: Examples of different grammatical structures containing reporters. On the left, the sentence in English. On the right, the parallel sentence in Spanish.

Example 1	
Remember the words of the expert, now Prime Minister Nikol Pashinyan, during the demonstrations: "The future of Armenia depends on [. . .]"	*Recuerda las palabras del experiodista, ahora Primer Ministro Nikol Pashinyan, durante las manifestaciones: "El futuro de Armenia depende de [. . .]"*
Example 2	
Facebook announced that it has deactivated 32 accounts and pages in its social network [. . .]	*Facebook anunció que ha desactivado 32 cuentas y páginas en su red social [. . .]*
Example 3	
According to a sentence provided by the High Court of Justice, the Court considers him guilty of the crimes of [. . .]	Según consta en una sentencia facilitada por el Tribunal Superior de Justicia, la Sala le considera culpable de los delitos de [. . .]
Example 4	
The head of Mosquito Alert's entomologist team, Roger Eritja, affirmed: "After reviewing the area [. . .]"	*El jefe del equipo de entomólogos de Mosquito Alert, Roger Eritja, ha afirmado: "Después de revisar la zona [. . .]"*
Example 5	
The Official State Gazette (OSG) has published this Saturday the penalty of almost 1.5 million euros [. . .]	*El Boletín Oficial del Estado (BOE) ha publicado este sábado la multa de casi 1,5 millones de euros [. . .]*
Example 6	
Both Finland and other states, such as Sweden, have publicly criticized Portuguese legislation.	*Tanto Finlandia como otros Estados, caso de Suecia, han hecho públicas sus críticas a la legislación portuguesa.*
Example 7	
The mosquito 'Aedes japonicus' has arrived for the first time in Spain and Southern Europe, according to the first report of Risk Rapid Assessment issued by the Coordination Centre for Health Alerts and Emergencies this July, fruit of the alert received from Asturias through the Mosquito Alert platform, has reported the Creaf this Wednesday in a statement.	*El mosquito 'Aedes japonicus' ha llegado por primera vez a España y al Sur de Europa, según revela el primer informe de Evaluación Rápida de Riesgo emitido por el Centro de Coordinación de Alertas y Emergencias Sanitarias este mes de julio, fruto de la alerta recibida desde Asturias a través de la plataforma Mosquito Alert, ha informado el Creaf este miércoles en un comunicado.*

As it can be seen with these few examples, the issue to face is labelling sequential text to extract the proper entity that provides the information. Taking into consideration the variety of sequences and grammatical structures that journalists can write, in this article we propose the application of Conditional Random Fields (CRFs) for this specific type of ER task for the Spanish language, namely, to identify what we have called the *"reporter"* in newspaper articles, that is, to spot who is the provider of the information.

Accordingly, the rest of the article is structured as follows: Section 2 presents some related work; Section 3 provides a brief introduction to the theoretical framework; Section 4 details all the information related to the experimental setup and results; finally, Section 5 provides some conclusions.

2. Related work

2.1. Natural Language Processing for journalism

NLP techniques have been widely used in newspapers. Thus, currently we can mention recent works on how several authors use NLP to perform tasks like NER [9, 10], automatic summarization [11, 12], automatic annotation of keywords [13] and subtopic [14], automatic deception detection [15], opinion mining [16, 17, 18, 19], text mining for knowledge extraction [20], predicting the relevance of posts in social media [21], automatic generation of headlines based on well-known expressions [22], identifying sensational episodes of news events [23], analysis of urban legends [24], etc.

In addition to the mentioned works, we highlight those performed within the topic of quoted extraction and attribution [25], which tries to assign the appropriate speaker to each quote, even though other kinds of information like assertions, beliefs, facts and eventualities [26] can be attributed.

Thus, in this regard, although they are initial approaches to the issue, [27] presents experiments in indirect and mixed quotation extraction and attribution using the four methods introduced by O'Keefe *et al.* [25], and [28] details a joint model for entity-level quotation attribution and coreference resolution.

More recently, [29] describes an approach that integrates event extraction with attribution extraction to identify individual accounts of events about industrial regeneration from news articles. Its authors perform the NER task using neural networks with CRF, and the event extraction using semantic role labelling (to identify whether the word acts as an agent, patient, etc.) and a lexicon of event nouns. Then, they use a lexicon of attribution verbs to detect whether a sentence conveys attribution. In the affirmative case, they analyse the dependency parse of the sentence to join the event to the corresponding agent if the verb is succeeded by a *that*-clause.

As seen above, in spite of the number of research works that use NLP in newspapers in some way, to the best of our knowledge, no work performs the extraction of those entities that provide the information to contribute to the journalism transparency and even less for the Spanish language.

2.2. Labelling sequential data

As previously introduced, the task of extracting from the text those entities that act as information providers can be considered as a labelling sequential data problem.

When labelling sequential data, Hidden Markov Models (HMMs) [30] are one of the most widely popular sequential models for information extraction, which is a generative model based on joint probability distributions. However, the use of HMMs is tied to processing linear-sequence observations.

Whether it is necessary to identify a sequence that can be arbitrarily structured, Conditional Random Fields (CRFs) appears as an alternative to the related HMM [31, 32, 33, 34]. CRFs are a stochastic statistical sequence modelling method that has been widely used in fields like Bioinformatics, Computer Vision, and NLP.

Within the NLP field, CRFs take the context (a sliding window of the neighbour words) into account to label a sequence of input words. To name some of the most popular tasks where this method has been applied in NLP, we can mention Part-Of-Speech Tagging (POS Tagging), Named Entity Recognition (NER) and shallow parsing for information extraction.

Neural Networks (NN) has burst in the field of NLP for a wide range of tasks, and sequencing labelling is not an exception. Thus, approaches like those based in Recurrent NN (RNN) or its variant known as Bidirectional Long Short-Term Memory (BiLSTM) [35, 36, 37, 38, 39, 40] have emerged as alternatives to CRFs, thanks to the fact that they allow capturing the sequential information due to their ability to use context when mapping between input and output sequences [41].

Nevertheless, despite the emergence application of NN for sequencing labelling and their performance, CRFs are currently still considered a state-of-the-art approach.

2.3. CRFs to extract relevant information from the text

Naming some examples of CRFs for information retrieval tasks, we can highlight the achievements by [42] for the shared task at CoNLL-2003[2] to perform NER for English and German languages.

For their part, going further NER, [7] and [8] use CRFs for ER in Bengali and Assamese languages respectively. Besides, [43] models the ER task using a CRFs layer jointly to the relation extraction task to potentially identify multiple relations for each entity.

In turn, [44] takes the identification of the sources of opinions, emotions and sentiments as an information extraction task, and thus, they use CRFs together with extraction patterns to perform it. For their part, not for information extraction but applied to a classification task, [45] proposes a method based on dependency trees using CRFs with hidden variables for sentiment classification of Japanese and English subjective sentences.

In this field of Opinion Mining, to analyze the relationship between the number of opinion targets and the sentiment expressed in that sentence, [46] uses BiLSTM with CRF (BiLSTM-CRF) and Convolutional Neural Networks (CNN). Particularly, the authors use the first layer with BiLSTM-CRF to classify the sentences as non-target, one-target or multi-target, depending on whether there are none, one or more targets in the opinion. Also, this BiLSTM-CRF layer performs the opinion targets extraction, i.e., to identify the entity on which an opinion has been expressed. In the second layer, they use CNN to perform the sentiment classification.

Similarly, viewing sentiment detection as a sequence labelling problem, [47] extracts jointly the entities and the sentiment expressed towards them. Its authors apply the approach using CRFs to build models for Spanish and English and use them on tweets. Likewise, using the data of [47], [48] analyzes the effect of

[2] `https://www.clips.uantwerpen.be/conll2003/ner/`

word embedding and automatic feature combinations by extending a CRFs baseline using neural networks for sentiment analysis.

Mining legal texts, [49] trains a linear-chain CRF to automatically recognize and extract those citations from legal documents. Following, the authors build a citation graph with automatically labelled edges according to whether they are a legal basis, a definition, an exception, etc.

All these reference works have provided interesting results regarding the use of CRFs to extract information from the text where the sequencing structure is arbitrary.

3. Theoretical base: Linear-chain CRF

According to [31, 32, 33, 34], a linear-chain CRF can be defined as the probability of a particular sequence y given the observation sequence x, i.e., a conditional distribution $p(y|x)$ as follows:

$$p(y|x) = \frac{1}{Z(x)} \prod_{t=1}^{T} exp \left\{ \sum_{k=1}^{K} \lambda_k f_k(y_{t-1}, y_t, x_t) \right\} \tag{1}$$

$$Z(x) = \sum_{y} \prod_{t=1}^{T} exp \left\{ \sum_{k=1}^{K} \lambda_k f_k(y_{t-1}, y_t, x_t) \right\} \tag{2}$$

where $Z(x)$ is a normalization function to provide a value in the range [0,1], T is the lenght of the sequence, K is the number of different features, f_k is the feature function to compute the $k-th$ feature, every λ_k is the weight for the f_k feature function, and y_{t-1}, y_t are the previous and the current positions in the label sequence respectively.

To avoid overfitting, the equations 1 and 2 use the λ_k parameters. Particularly, these λ_k parameters suppose a penalty on weight vectors as a regularization mechanism to avoid overfitting. The fine-tuning of these parameters could contribute to improving model performance. Therefore, during the training stage, it is necessary to find those λ_k parameters that best fit the training data.

To achieve this goal, we can use $L1$ and $L2$ regularization terms in optimization algorithms like Gradient Descent using the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) [50] or the Stochastic Gradient Descent with L2 regularization (L2SGD) [51]. These algorithms compute the gradient of the objective function to maximize the logarithm of the likelihood of the training data.

In these optimization algorithms, the $L1$ represents a Least Absolute Shrinkage and Selection Operator (LASSO) regularization, and $L2$ supposes a Ridge regularization. This way, with the first one, we reduce the less important features coefficient to zero, removing those features that are less relevant, and thus, providing a way to select features if we have a lot of them. With the second one, the algorithm is able to smooth the values in order to avoid the complexity of the models.

However, although L-BFGS is the most widely used optimization algorithm in CRFs because it provides $L1$ and $L2$ regularization, and SGD supposes a good alternative to applying $L2$ regularization, other algorithms can be used to compute the feature weights. Particularly, we can mention the Averaged Perceptron (AP) [52], which uses the average of feature weights, Passive Aggressive (PA) [53], which adapts the weights to adjust data to a new distribution, only if detects that data comes from a completely different distribution, or the Adaptive Regularization Of Weight Vector (AROW) [54], which initializes the vector of feature weights as a multivariate Gaussian distribution.

4. Experimental setup

This section will provide the details about the experimental setup: identifying the classes of sequence, describing the built corpus, describing the features, defining the metrics to measure the performance, specifying the steps performed in the experimentation process, and finally, analyzing the obtained results.

4.1. Classes of sequence to identify

To perform the experiment, we collaborated with Público[3], a Spanish online newspaper. After an interview with the journalists in the redaction, we decided to categorize the reporters as follows:

- Location (LOC), what journalists use to refer to the regional government or the people of a specific geographical area.
- Media (MED), used when journalists indicate that the information has been provided by another media or news agency.
- Organizations (ORG), when the information comes from governmental or non-governmental entities, political parties, companies, etc.
- Persons (PER), to identify specific person names, but excluding their roles, like *"Prime Minister"* or *"Chief Executive Officer"*.
- Miscellaneous (MISC), to identify any other reporters that can not be included in any the previous classes, like Laws, books, reports, etc.

It is interesting to notice that, despite this classification, while checking the labelled sentences in the newspaper articles, we identified that journalists use the *"personification"* imperative figure of speech. In brief, journalists use to personify reporters like organizations, companies, etc. As a consequence, they syntactically use the same grammatical structures in sentences, independently of the kind of reporter. This

[3] `https://www.publico.es`

way, it is easy to find sentences like *"Spanish Supreme Court affirms that decisions of UN treaty bodies are [...]"*, where *"Spanish Supreme Court"* works syntactically the same way a proper name of a person. The reason for this is that journalists look not interested in determining the kind of reporter, that is, they just wanted to know who (the person) or what (the organization, the company, the report, etc.) provides the information but not its type.

This leads us to guess that the type of reporter may not be necessary and that only one class could be significant, namely, to label only the sequence *"reporter"* (R). Thus, both a multi-class-of-sequence approach and a one-class-of-sequence approach have been explored.

4.2. Corpus description

For the experimental setup, the first step is to build the labelled corpus. Particularly, our corpus contains 604 newspaper articles in Spanish. They were gathered from August 2018 to August 2019 from the Público site, an online Spanish newspaper. In these set of articles, we have manually labelled up to 1669 sentences as be written in both, direct speech or reported speech. For each of these labelled sentences, we also identified and tagged the reporter, i.e., *who says that*, in the sentence. Table 2 details the statistics with the labelled data contained in the corpus.

Total newspaper articles	604
Total sentences containing reporter	1669
Total labelled entities	1903
labelled as Location (LOC)	11
labelled as Media (MED)	185
labelled as Organization (ORG)	593
labelled as Person (PER)	1016
labelled as Miscellaneous (MISC)	103
Average of tokens per labelled sentence	45.11 (17.02)
Average of tokens per entity	2.31 (1.82)

Table 2: Corpus statistics for the labelled newspaper articles

We store news in an XML file like in listing 1. This XML keeps the structure of the paragraphs of the original news with the whole text, to allow future analysis and better processing. For instance, the reporter may have been indicated not necessary in the same sentence but another in the same or different paragraph. Thus, as the listing shows, every news input has an URL to its online version, as well as its paragraphs (each one tagged as *"p"*). Within the paragraph, whether a sentence contains a reporter that provides some kind of information, then this sentence is tagged as *"report"* and the reporter is tagged as *"reporter"* with an

```xml
<news_article url='http://www.publico.es/sociedad/insectos-llega-espana-nuevo-
    mosquito-invasor-origen-asiatico.html'>

  <p>
    <report>El mosquito 'Aedes japonicus' ha llegado por primera vez a Espana y al
         Sur de Europa, según revela el primer informe de Evaluación Rápida de
        Riesgo emitido por el <reporter type="ORG">Centro de Coordinación de
        Alertas y Emergencias Sanitarias</reporter> este mes de julio, fruto de la
        alerta recibida desde Asturias a través de la plataforma Mosquito Alert, ha
         informado el <reporter type="ORG">Creaf</reporter> este miércoles en un
        comunicado.</report>
  </p>
...
  <p>
    <report>El jefe del equipo de entomólogos de Mosquito Alert, <reporter type="
        PER">Roger Eritja</reporter>, ha afirmado: "Después de revisar la zona
        hemos podido encontrar todas las fases biológicas del vector en varios
        puntos alejados entre sí, lo que sugiere que el mosquito está ya
        establecido en un área que puede ser mucho más amplia, aunque se necesitará
        n más estudios para confirmarlo".</report> La mayor preocupación de la
        llegada del mosquito Aedes japonicus es que, aparte de causar molestias con
         sus picaduras similares a las de los demás mosquitos, tiene la capacidad
        de transmitir varios virus entre los cuales el más relevante en Espana serí
        a el del Nilo Occidental.
  </p>
...
</news_article>
```

Listing 1: Snippet of an labelled newspaper article in XML

attribute that indicates its type (a person, an organization, a media, a location, or miscellaneous, according to section 4.1).

With this XML format, it is easy to gather those sentences tagged as *"report"* and transform them to an IOB labelling model. As a result, the sentences are represented in the way shown in listing 2. In this listing, we can see how each word and punctuation mark of the sentence is labelled. According to the IOB model, if the word is part of an entity, it is labelled with its type (ORG for organization in this case) and the prefix that indicates whether it is at the beginning (B-) of the chunk, inside (I-) of the chunk, or outside (O) of the chunk.

4.3. Features selection

In addition to the word itself in lowercase, we have identified two groups of features, namely, lexical and syntactical features. With the lexical features group, we discover clues about the word, taking into account how it has been written, i.e. its form. With the syntactical features group, we look for clues about the function the word has in the sentence and its relations with other words.

Lexical features. This kind of features is oriented to identify relevant characteristics of the words form, like if they were written in titlecase (what may indicate that they are a proper noun), whether they were written

```
...
según O
revela O
el O
primer O
informe O
de O
Evaluación O
Rápida O
de O
Riesgo O
emitido O
por O
el O
Centro B-ORG
de I-ORG
Coordinación I-ORG
de I-ORG
Alertas I-ORG
y I-ORG
Emergencias I-ORG
Sanitarias I-ORG
este O
mes O
de O
julio O
...
```

Listing 2: IOB representation for a labelled sentence

all in uppercase (what may indicate that it is a company name or an acronym), if they contain dots and slash (indicating that they could be abbreviations), etc.

In particular, we selected the next list of lexical feature functions:

- Word-case features, particularly: *is_uppercase*, *is_titlecase* and *is_digit*.
- Lemma of the word, to remove the possible conjugation, pluralization, etc. that the word has suffered.
- Suffixes, more specifically: the three and the two last letters or the word.
- Punctuation ratio defined as: $\frac{|\{x\} \cap \{`.`, `-`\}|}{length(x)}$, where $\{x\}$ are the letters of the word x
- Vowels ratio defined as: $\frac{|\{x\} \cap \{a, e, i, o, u\}|}{length(x)}$, where $\{x\}$ are the letters of the word x

With the previous lexical features, we will be able to identify special words that are suitable as names for entities or whether they are regular words from the vocabulary. In particular, if all the letters from a word are uppercased, it can be a clue for identifying acronyms (even more in the case of a lack or an excess of vowels measured by the vowel ratio), a titlecased word can indicate a proper name, the usage of a lot of punctuation marks (measured by the punctuation ratio) can point out we have found abbreviations, the use of specific suffixes can designate particular forms and functions of the words (whether they are acting as an adverb, adjective, substantive, ...), etc.

Syntactical features. This set of features provides information about the kind of word and its function within the sentence. Grammatical classes (nouns, adjectives, verbs, etc.) are particularly important, and they can be extracted using a POS tagger.

Specifically, we selected the next list of syntactical features:

- POS tag indicating if it is a noun (singular or plural), an adjective (personal or possessive), a verb (in base form, past tense, ...), etc.[4]
- First two characters of the POS tag of the word, i.e., the kind of word *without* indicating if it is a plural or singular noun, a personal or possessive pronoun, a comparative or possessive adjective, etc. Unlike the previous one, this feature only indicates the function of the word in the sentence, but it does not go into more detail.
- Role of the word in the sentence, i.e., subject, main verb, etc.
- Related verb whether available. In sentences containing transitive or intransitive verbs, the verbs are closely related to the direct or indirect object in the active voice, or the subjects in the passive voice. *"say"*, *"affirm"*, etc. are transitive verbs, and they can provide useful information on who does the action. That is, it could be representative to link a verb like *"affirm"* to its specific subject.

4.4. Performance measurement

We have used *precision*, *recall* and $f1$-score to measure the performance of the classifier per class of sequence at sentence level, and consequently to identify what classes of sequence better performs. Using micro and macro averages to aggregate these metrics will provide us with the classifier performance, i.e., aggregation of the obtained values including all classes of sequence (micro average), against aggregating of the average computed independently for each class of sequence (macro average).

In adition, we have computed the sequence *accuracy* (i.e. exact match ratio) taking into account matches only when two sequences are equal in the validation and the classifier prediction, i.e. to compute exact matching at the sequence level.

4.5. Baseline

To help us to estimate how good are the obtained results, we established a baseline based on the next heuristic: if the sentence contains a reporting verb (like *"say"*, *"tell"* or *"affirm"*) or an *"according to"*

[4]The whole tag set can be found listed in `https://www.clips.uantwerpen.be/pages/mbsp-tags`

expression (*“según”* in Spanish), it indicates the use of direct or reported speech, and then we will extract those named entities (person, organization, location or miscellaneous) that act as subject or object in the sentence. To build the lexicon of reporting verbs, we collected all of them that appear in the dataset. This baseline provides a multi-class-of-sequence approach depending on whether the entities are PER, ORG, LOC or MISC in both direct or reported written sentences.

4.6. Software and tools

To perform the experiment, we used a Part-Of-Speech (POS) tagger that helps us to compute some of the input features and a CRFs implementation to create different CRFs models. To analyse the performance of the models, we used a framework designed to evaluate labelling sequences results and a NER tool for implementing the defined baseline.

Part-of-speech tagger. To perform the POS-Tagging for the Spanish language, we chose pattern.es [55], a Python library which provides a fast POS tagger for Spanish as well as verb conjugation and noun singularization and pluralization.

CRFs Implementation. To build and test the CRFs model, we selected the implementation of CRFsuite [56], which provides fast training and tagging algorithms relying on libraries like libLBFGS [57] for numerical optimization. More specifically, we used the Python binding of CRFsuite [58] that allows compatibility with scikit-learn using a thin wrapper [59].

Permormance measuring. Because we use the scikit-learn wrapper for CRFsuite, we can apply the sklearn interface for multilabel problems performance measuring. However, since our problem consists of labelling sequences, we will use the seqeval [60] python-based framework. This framework is based on the well-tested and widely accepted Perl script conlleval designed to evaluate the results of processing the CoNLL-2000 shared task.

Named Entity Recognition tool. For the implementation of the defined baseline that will allow us to compare the results of our approach, we will use the named entity recognition tool included in spacy [61] because it allows labelling sequences as PER, LOC, ORG and MISC for the Spanish language, exactly as we defined in section 4.1.

4.6.1. CRFs setup

There are several issues to take into account and some parameters that we must to finetuning for the experimental setup. In this section, we will provide the details we used in our experimental setup.

The first issue is the context, i.e., the word sliding window. In our setups, we defined sliding windows with values three and five. Thus, for every word from the text (but the first and the last one), we process its own features and the same for previous (or two previous) and next (or two next).

```
...
[ # list of features for word 'de'
'word.lower=de',                                    # the word in lowercase
'word[-3:]=de', 'word[-2:]=de',                     # the 3 and 2 last letters
'word.isupper=False', 'word.istitle=False',
'word.isdigit=False',                               # word-case features
'word.punctratio=0.0', 'word.vowelsratio=0.0',      # ratios
'postag=IN', u'postag[:2]=IN',                      # POS tag features
'role=NoRole',                                      # role in the sentece
'word.lemma=de',                                    # lemma
'verb=explicar',                                    # related verb

# same as before but for the previous word in the sentence
'-1:word.lower=época', '-1:word[-3:]=oca', '-1:word[-2:]=ca',
'-1:word.isupper=False', '-1:word.istitle=False', '-1:word.isdigit=False',
'-1:word.punctratio=0.0', '-1:word.vowelsratio=0.0',
'-1:postag=NN', '-1:postag[:2]=NN',
'-1:role=NoRole', '-1:word.lemma=época', '-1:verb=explicar',

# same as before but for the next word in the sentence
'+1:word.lower=serge', '+1:word[-3:]=rge', u'+1:word[-2:]=ge',
'+1:word.isupper=False', '+1:word.istitle=True', '+1:word.isdigit=False',
'+1:word.punctratio=0.0', '+1:word.vowelsratio=0.0',
'+1:postag=NNP', '+1:postag[:2]=NN',
'+1:role=NoRole', '+1:word.lemma=serge', '+1:verb=explicar'
],

[ # list of features for word 'Serge'
'word.lower=serge', 'word[-3:]=rge', 'word[-2:]=ge',
'word.isupper=False', 'word.istitle=True', 'word.isdigit=False',
'word.punctratio=0.0', 'word.vowelsratio=0.0',
'postag=NNP', 'postag[:2]=NN',
'role=NoRole', 'word.lemma=serge', 'verb=explicar',

'-1:word.lower=de', '-1:word[-3:]=de', '-1:word[-2:]=de',
'-1:word.isupper=False', '-1:word.istitle=False', '-1:word.isdigit=False',
'-1:word.punctratio=0.0', '-1:word.vowelsratio=0.0',
'-1:postag=IN', '-1:postag[:2]=IN',
'-1:role=NoRole', '-1:word.lemma=de', '-1:verb=explicar',

'+1:word.lower=sargsián', u'+1:word[-3:]=ián', u'+1:word[-2:]=án',
'+1:word.isupper=False', '+1:word.istitle=False', '+1:word.isdigit=False',
'+1:word.punctratio=0.0', '+1:word.vowelsratio=0.0',
'+1:postag=NN', '+1:postag[:2]=NN',
'+1:role=NoRole', '+1:word.lemma=sargsián', '+1:verb=explicar'
]
...
```

Listing 3: Features representation for words *"de Sege"* in *"[. . .] época de Serge Sargsián."*

The second issue to consider is how to manage the numerical ratios of the features. Although CRFs itself can manage numerical features, the CRFSuite API does not support adding float features. The only way this suite provides to support float features is by mapping key-string labels to float values. Therefore, ratio values are rounded to one decimal point and converted to a string, limiting the possible values to those from the list ["0.0", "0.1", "0.2", . . . , "1.0"]

Accordingly, and following the instruction of the CRFSuite API, we build the list of features in the way shown in listing 3 for every word in the text. As we can see, we computed the list of key-string pairs (coded as a string *'key=value'*) for every feature of the word. The list of features for every word includes those features of the words that are in its sliding window. A prefix with a number (-2, -1, +1, +2, etc.) is used to identify if the feature corresponds to the previous word, the next one, and so on.

4.7. Features selection

The third point to keep in mind is the way to fine-tuning the λ_k parameters (the penalty on weight vectors) to improving the performance. It is needed an optimization algorithm that computes the gradient of the objective function. To achieve that, CRFsuite implements a complete list of training algorithms [56], namely: L-BFGS, L2SGD, AP, PA and AROW (see section 3). In the design of our experimental setup, we look for a set of possible combinations to try to cover a spectrum that allows us to draw some conclusions.

4.8. Performing the experiment

Figure 1 details the steps we performed. As the Figure shows, we start loading the XML file and fetching those sentences tagged as *"report"*, and that we transform to IOB format to define the proper token sequence and the labelling to work with (section 4.2). After that, we compute all the features for every token (section 4.3). Later, we perform a *k-fold* cross-validation with 3-folds for all CRFs configurations we selected and implemented (section 4.6). The cross-validation analysis results will allow us to identify the best CRFs configuration in order go deeper to analyze that classifier, computing the performance per class of sequence with a train-test split, analyzing the configuration performance (section 5) and comparing them with the corresponding baseline.

5. Results

Following the plan of activities (see Figure 1), after taking the labelled sentences from our newspaper article corpus, converting them to IOB format, and computing the features, we performed 3-folds cross-validation over a total of 44 different CRFs setups using a computer with an Intel(R) Core(TM) i3-4005U CPU 1.70GHz and 8Gb of RAM. Tables 3 and 4 details all the CRFs setup we performed using a 3-tokens and 5-tokens sliding windows respectively. In the first two columns, we can see the kind of training algorithm and its parameters to customize. Thereafter, we find the Mean and Standard Deviation (in parentheses) for the *precision*, *recall*, $f1$-score metrics, as well as for the score time and fit time for each CRFs configuration.

In Tables 3 and 4, the most obvious result is that the fitting times are higher the more complex is the algorithm parametrization. This is particularly remarkable in the configurations defined for L-BFGS. However, despite the time consuming of some of these configurations, taking into account the stochastic nature of CRFs, these differences in performance are not really outstanding.

Similarly, comparing the results between these both tables, the higher is the sliding window, the higher are the fitting times, but the increase of the performance looks not really remarkable.

The Tables also highlight the best $f1$-score for each training algorithm to compare them. We can observe that the use of AROW as training algorithm provides the worst results. For the rest of the algorithms, using $f1$-score as the precision metric of the classifiers, we must study the second and third decimal point in most

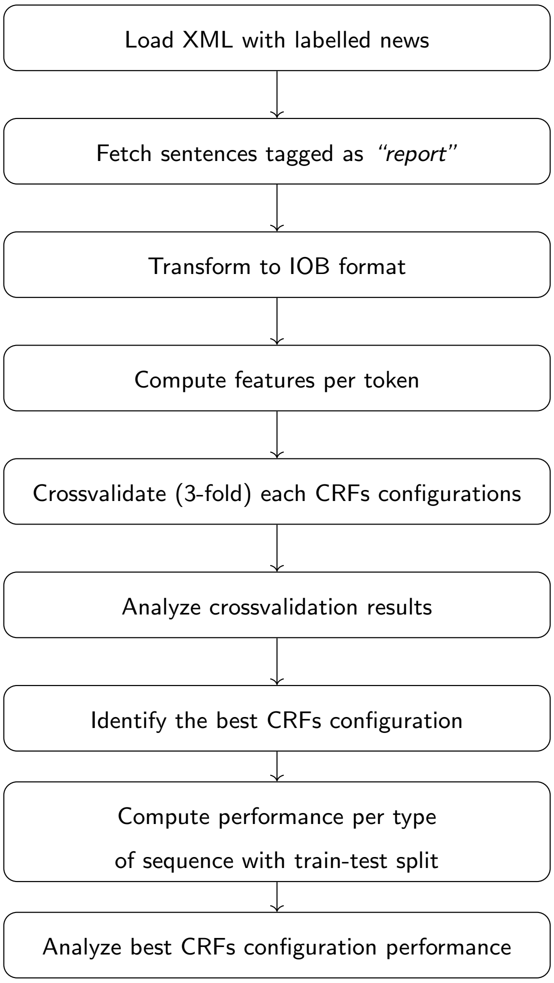

Figure 1: Plan of activities performed in the experimentation process.

algorithm	cfg	parameters	precision	recall	f1-score	score time	fit time
AROW	01	(AST=False)	0.488 (0.054)	0.489 (0.066)	0.487 (0.056)	6.414 (0.836)	18.542 (1.621)
AROW	02	(AST=True)	0.492 (0.039)	0.519 (0.059)	0.505 (0.047)	6.754 (1.116)	28.505 (1.919)
AROW	03	(AST=True; VAR=0.5)	0.540 (0.054)	0.528 (0.077)	0.533 (0.064)	6.215 (0.514)	28.522 (2.602)
AROW	04	(AST=True; VAR=0.25)	0.574 (0.032)	0.539 (0.065)	**0.555 (0.048)**	6.239 (0.442)	28.300 (2.405)
AP	01	(AST=False)	0.714 (0.036)	0.592 (0.036)	0.646 (0.031)	6.949 (1.517)	20.077 (1.641)
AP	02	(AST=True)	0.701 (0.037)	0.598 (0.043)	**0.645 (0.036)**	6.843 (1.056)	31.943 (3.365)
PA	01	(AST=False; PA type I)	0.727 (0.025)	0.600 (0.032)	0.657 (0.022)	6.064 (0.461)	20.897 (1.765)
PA	02	(AST=True; PA type I)	0.722 (0.030)	0.615 (0.032)	0.664 (0.026)	7.137 (1.725)	32.263 (3.310)
PA	03	(AST=False; PA without slack variables)	0.727 (0.025)	0.600 (0.032)	0.657 (0.022)	6.171 (0.339)	20.615 (1.640)
PA	04	(AST=True; PA without slack variables)	0.722 (0.030)	0.615 (0.032)	0.664 (0.026)	7.172 (1.636)	33.385 (3.718)
PA	05	(AST=False; PA type II)	0.728 (0.031)	0.601 (0.035)	0.657 (0.026)	6.540 (1.123)	20.183 (1.843)
PA	06	(AST=True; PA type II)	0.724 (0.025)	0.616 (0.025)	**0.665 (0.021)**	7.313 (1.633)	35.281 (4.517)
PA	07	(AST=False; PA type I; c=0.5)	0.727 (0.025)	0.600 (0.032)	0.657 (0.022)	6.041 (0.416)	20.214 (1.636)
PA	08	(AST=True; PA type I; c=0.5)	0.722 (0.030)	0.615 (0.032)	0.664 (0.026)	6.430 (0.536)	31.330 (2.632)
PA	09	(AST=False; PA type II; c=0.5)	0.727 (0.032)	0.602 (0.029)	0.658 (0.023)	6.316 (0.721)	19.454 (1.371)
PA	10	(AST=True; PA type II; c=0.5)	0.722 (0.034)	0.610 (0.026)	0.661 (0.024)	7.119 (1.420)	32.244 (2.979)
L2SGD	01	(AST=False; c2=1.0)	0.768 (0.028)	0.571 (0.069)	0.653 (0.050)	5.997 (0.280)	49.459 (10.769)
L2SGD	02	(AST=True; c2=1.0)	0.775 (0.032)	0.563 (0.055)	0.650 (0.042)	6.432 (0.254)	71.847 (11.916)
L2SGD	03	(AST=True; c2=0.01)	0.745 (0.042)	0.618 (0.018)	0.675 (0.024)	7.297 (1.478)	75.000 (41.544)
L2SGD	04	(AST=True; c2=0.02)	0.746 (0.042)	0.618 (0.018)	**0.676 (0.024)**	7.001 (0.878)	73.451 (39.014)
L2SGD	05	(AST=True; c2=0.05)	0.750 (0.045)	0.612 (0.015)	0.673 (0.021)	7.386 (1.525)	67.514 (27.466)
L2SGD	06	(AST=True; c2=0.1)	0.747 (0.041)	0.616 (0.019)	0.675 (0.024)	6.836 (0.707)	65.478 (32.904)
L2SGD	07	(AST=True; c2=0.2)	0.751 (0.044)	0.604 (0.013)	0.669 (0.018)	7.558 (1.815)	63.727 (22.749)
L-BFGS	01	(AST=False; c1=0.0; c2=1.0)	0.774 (0.033)	0.560 (0.044)	0.649 (0.035)	6.141 (0.407)	114.622 (23.121)
L-BFGS	02	(AST=True; c1=0.0; c2=1.0; LS=MT)	0.776 (0.035)	0.571 (0.040)	0.657 (0.031)	6.751 (0.528)	155.986 (16.464)
L-BFGS	03	(AST=True; c1=0.0; c2=1.0; LS=BT)	0.777 (0.036)	0.571 (0.040)	0.657 (0.031)	6.734 (0.534)	158.492 (21.703)
L-BFGS	04	(AST=True; c1=0.0; c2=1.0; LS=SBT)	0.777 (0.036)	0.571 (0.040)	0.657 (0.032)	6.919 (0.876)	180.232 (26.515)
L-BFGS	05	(AST=True; c1=0.0; c2=0.01; LS=MT)	0.731 (0.027)	0.593 (0.042)	0.654 (0.034)	7.962 (2.108)	409.419 (44.826)
L-BFGS	06	(AST=True; c1=0.0; c2=0.02; LS=MT)	0.741 (0.029)	0.594 (0.037)	0.659 (0.031)	6.688 (0.655)	373.536 (32.855)
L-BFGS	07	(AST=True; c1=0.0; c2=0.05; LS=MT)	0.752 (0.028)	0.595 (0.035)	0.664 (0.027)	6.781 (0.687)	320.469 (15.008)
L-BFGS	08	(AST=True; c1=0.0; c2=0.1; LS=MT)	0.762 (0.029)	0.596 (0.034)	0.668 (0.024)	6.633 (0.406)	285.131 (48.634)
L-BFGS	09	(AST=True; c1=0.0; c2=0.1; LS=BT)	0.762 (0.030)	0.596 (0.034)	0.668 (0.025)	7.994 (2.269)	265.536 (41.729)
L-BFGS	10	(AST=True; c1=0.0; c2=0.2; LS=MT)	0.771 (0.032)	0.593 (0.040)	0.669 (0.029)	6.740 (0.568)	234.821 (34.217)
L-BFGS	11	(AST=True; c1=0.0; c2=0.3; LS=MT)	0.771 (0.031)	0.590 (0.040)	0.668 (0.030)	7.813 (2.257)	223.509 (36.444)
L-BFGS	12	(AST=True; c1=0.0; c2=0.01; LS=MT)	0.778 (0.035)	0.569 (0.040)	0.656 (0.031)	6.270 (0.366)	1632.211 (231.505)
L-BFGS	13	(AST=True; c1=0.0; c2=0.02; LS=MT)	0.779 (0.036)	0.570 (0.039)	0.657 (0.031)	6.178 (0.295)	1437.545 (286.828)
L-BFGS	14	(AST=True; c1=0.0; c2=0.05; LS=MT)	0.779 (0.035)	0.568 (0.035)	0.656 (0.029)	6.117 (0.405)	1564.135 (72.671)
L-BFGS	15	(AST=True; c1=0.1; c2=1.0; LS=MT)	0.783 (0.039)	0.567 (0.035)	0.657 (0.030)	6.093 (0.346)	1623.002 (209.590)
L-BFGS	16	(AST=True; c1=0.2; c2=1.0; LS=MT)	0.781 (0.036)	0.563 (0.039)	0.653 (0.033)	7.086 (1.959)	1699.284 (158.537)
L-BFGS	17	(AST=True; c1=0.3; c2=1.0; LS=MT)	0.779 (0.039)	0.559 (0.037)	0.650 (0.033)	5.818 (0.361)	1579.708 (190.997)
L-BFGS	18	(AST=True; c1=0.1, c2=0.1; LS=MT)	0.764 (0.025)	0.598 (0.034)	0.671 (0.027)	5.695 (0.247)	2289.187 (439.256)
L-BFGS	19	(AST=True; c1=0.1, c2=0.2; LS=MT)	0.765 (0.028)	0.596 (0.032)	0.670 (0.026)	5.797 (0.332)	2164.879 (201.293)
L-BFGS	20	(AST=True; c1=0.2, c2=0.1; LS=MT)	0.764 (0.027)	0.603 (0.033)	**0.673 (0.028)**	6.611 (1.722)	1968.946 (309.368)
L-BFGS	21	(AST=True; c1=0.2, c2=0.2; LS=MT)	0.766 (0.029)	0.597 (0.034)	0.671 (0.029)	5.679 (0.334)	2296.530 (334.906)

Table 3: Crossvalidation results training classifiers for all the classes of sequences using a 3-tokens sliding window. Mean and standard deviation in parentheses for the precision, recall, $f1$-score, score time and fit time for each CRFs configuration. AST=all possible states and transitions; c1=coeficient for $L1$ regularization; c2=coeficient for $L2$ regularization; LS=linesearch method (MT=More and Thuente, BT=Backtracking, SBT=Strong Backtracking); c=aggressiveness parameter used for PA-I and PA-II (controls the influence of the slack term on the objective function); VAR=variance

algorithm	cfg	parameters	precision	recall	f1-score	score time	fit time
AROW	01	(AST=False)	0.502 (0.052)	0.493 (0.059)	0.497 (0.056)	9.979 (0.750)	28.399 (1.071)
AROW	02	(AST=True)	0.550 (0.060)	0.533 (0.082)	0.541 (0.071)	10.874 (0.490)	49.858 (3.253)
AROW	03	(AST=True; VAR=0.5)	0.594 (0.036)	0.555 (0.063)	0.573 (0.050)	10.381 (0.758)	48.906 (4.358)
AROW	04	(AST=True; VAR=0.25)	0.612 (0.020)	0.564 (0.055)	**0.586 (0.040)**	10.935 (1.846)	50.400 (5.438)
AP	01	(AST=False)	0.716 (0.028)	0.603 (0.055)	0.654 (0.042)	10.102 (0.828)	32.672 (2.264)
AP	02	(AST=True)	0.728 (0.029)	0.621 (0.052)	**0.669 (0.038)**	10.630 (0.969)	52.622 (5.812)
PA	01	(AST=False; PA type I)	0.755 (0.029)	0.625 (0.040)	0.683 (0.033)	10.371 (0.657)	33.647 (2.374)
PA	02	(AST=True; PA type I)	0.743 (0.030)	0.630 (0.035)	0.681 (0.028)	12.314 (3.077)	58.881 (7.361)
PA	03	(AST=False; PA without slack variables)	0.755 (0.029)	0.625 (0.040)	0.683 (0.033)	10.335 (0.734)	33.794 (2.590)
PA	04	(AST=True; PA without slack variables)	0.743 (0.030)	0.630 (0.035)	0.681 (0.028)	10.834 (0.939)	54.587 (5.702)
PA	05	(AST=False; PA type II)	0.745 (0.027)	0.616 (0.045)	0.674 (0.035)	10.547 (1.151)	34.344 (2.649)
PA	06	(AST=True; PA type II)	0.747 (0.024)	0.632 (0.040)	**0.684 (0.030)**	12.041 (2.718)	60.569 (7.909)
PA	07	(AST=False; PA type I; c=0.5)	0.755 (0.029)	0.625 (0.040)	0.683 (0.033)	10.664 (1.371)	33.906 (2.647)
PA	08	(AST=True; PA type I; c=0.5)	0.743 (0.030)	0.630 (0.035)	0.681 (0.028)	11.937 (2.166)	55.301 (5.304)
PA	09	(AST=False; PA type II; c=0.5)	0.760 (0.032)	0.622 (0.044)	0.684 (0.037)	11.345 (2.320)	35.838 (3.010)
PA	10	(AST=True; PA type II; c=0.5)	0.744 (0.026)	0.629 (0.043)	0.681 (0.033)	11.912 (2.526)	52.611 (4.078)
L2SGD	01	(AST=False; c2=1.0)	0.798 (0.037)	0.577 (0.063)	0.668 (0.050)	10.237 (0.724)	63.574 (12.469)
L2SGD	02	(AST=True; c2=1.0)	0.770 (0.043)	0.617 (0.043)	0.685 (0.042)	12.702 (2.999)	102.471 (16.846)
L2SGD	03	(AST=True; c2=0.01)	0.771 (0.037)	0.617 (0.045)	0.685 (0.041)	13.057 (3.407)	105.278 (23.304)
L2SGD	04	(AST=True; c2=0.02)	0.771 (0.037)	0.617 (0.045)	0.685 (0.041)	11.574 (1.333)	96.284 (16.119)
L2SGD	05	(AST=True; c2=0.05)	0.772 (0.037)	0.618 (0.045)	0.686 (0.041)	12.917 (3.298)	98.796 (8.996)
L2SGD	06	(AST=True; c2=0.1)	0.771 (0.036)	0.619 (0.045)	0.686 (0.040)	12.856 (3.293)	102.181 (22.194)
L2SGD	07	(AST=True; c2=0.2)	0.774 (0.034)	0.619 (0.047)	**0.687 (0.041)**	11.971 (2.070)	122.860 (28.213)
L-BFGS	01	(AST=False; c1=0.0; c2=1.0)	0.798 (0.033)	0.584 (0.056)	0.673 (0.044)	11.334 (2.377)	187.223 (18.534)
L-BFGS	02	(AST=True; c1=0.0; c2=1.0; LS=MT)	0.796 (0.032)	0.597 (0.052)	0.681 (0.041)	13.592 (3.469)	299.761 (41.827)
L-BFGS	03	(AST=True; c1=0.0; c2=1.0; LS=BT)	0.796 (0.032)	0.596 (0.052)	0.681 (0.041)	11.897 (2.037)	279.252 (28.753)
L-BFGS	04	(AST=True; c1=0.0; c2=1.0; LS=SBT)	0.796 (0.032)	0.596 (0.052)	0.681 (0.041)	11.494 (1.236)	266.487 (35.129)
L-BFGS	05	(AST=True; c1=0.0; c2=0.01; LS=MT)	0.773 (0.027)	0.626 (0.043)	**0.691 (0.033)**	11.327 (1.020)	601.993 (78.486)
L-BFGS	06	(AST=True; c1=0.0; c2=0.02; LS=MT)	0.774 (0.025)	0.622 (0.043)	0.690 (0.033)	11.707 (1.522)	528.132 (54.927)
L-BFGS	07	(AST=True; c1=0.0; c2=0.05; LS=MT)	0.777 (0.029)	0.619 (0.043)	0.688 (0.035)	11.281 (1.126)	476.950 (54.153)
L-BFGS	08	(AST=True; c1=0.0; c2=0.1; LS=MT)	0.780 (0.031)	0.618 (0.044)	0.689 (0.035)	13.389 (3.837)	382.781 (50.540)
L-BFGS	09	(AST=True; c1=0.0; c2=0.1; LS=BT)	0.780 (0.031)	0.618 (0.044)	0.689 (0.035)	11.438 (1.098)	448.391 (50.949)
L-BFGS	10	(AST=True; c1=0.0; c2=0.2; LS=MT)	0.783 (0.030)	0.616 (0.045)	0.689 (0.035)	11.209 (0.970)	348.376 (45.719)
L-BFGS	11	(AST=True; c1=0.0; c2=0.3; LS=MT)	0.785 (0.030)	0.611 (0.049)	0.686 (0.039)	12.780 (3.441)	343.254 (30.789)
L-BFGS	12	(AST=True; c1=0.0; c2=0.01; LS=MT)	0.793 (0.033)	0.598 (0.050)	0.681 (0.040)	10.305 (0.612)	2630.096 (250.304)
L-BFGS	13	(AST=True; c1=0.0; c2=0.02; LS=MT)	0.791 (0.034)	0.597 (0.052)	0.680 (0.042)	10.335 (0.728)	2691.485 (319.258)
L-BFGS	14	(AST=True; c1=0.0; c2=0.05; LS=MT)	0.791 (0.034)	0.598 (0.050)	0.680 (0.040)	10.201 (0.747)	2656.542 (207.251)
L-BFGS	15	(AST=True; c1=0.1; c2=1.0; LS=MT)	0.787 (0.040)	0.592 (0.053)	0.674 (0.044)	9.966 (0.546)	2685.123 (374.923)
L-BFGS	16	(AST=True; c1=0.2; c2=1.0; LS=MT)	0.788 (0.038)	0.591 (0.053)	0.675 (0.044)	9.601 (0.658)	2690.847 (276.495)
L-BFGS	17	(AST=True; c1=0.3; c2=1.0; LS=MT)	0.787 (0.039)	0.586 (0.052)	0.671 (0.045)	9.629 (0.507)	2557.339 (477.647)
L-BFGS	18	(AST=True; c1=0.1, c2=0.1; LS=MT)	0.773 (0.027)	0.616 (0.048)	0.685 (0.038)	9.320 (0.525)	4673.357 (223.499)
L-BFGS	19	(AST=True; c1=0.1, c2=0.2; LS=MT)	0.779 (0.026)	0.612 (0.052)	0.685 (0.039)	9.568 (0.694)	4103.728 (435.075)
L-BFGS	20	(AST=True; c1=0.2, c2=0.1; LS=MT)	0.766 (0.027)	0.611 (0.047)	0.680 (0.037)	9.218 (0.578)	5034.042 (245.894)
L-BFGS	21	(AST=True; c1=0.2, c2=0.2; LS=MT)	0.777 (0.025)	0.610 (0.049)	0.683 (0.038)	11.294 (3.367)	3933.059 (623.774)

Table 4: Crossvalidation results training classifiers for all the classes of sequences using a 5-tokens sliding window. Mean and standard deviation in parentheses for the precision, recall, $f1$-score, score time and fit time for each CRFs configuration. AST=all possible states and transitions; c1=coeficient for $L1$ regularization; c2=coeficient for $L2$ regularization; LS=linesearch method (MT=More and Thuente, BT=Backtracking, SBT=Strong Backtracking); c=aggressiveness parameter used for PA-I and PA-II (controls the influence of the slack term on the objective function); VAR=variance

configurations to appreciate differences. In general, when we set the option to compute all possible states and transitions, we obtain slightly better results.

As we can see, hyperparameter tuning does not provide significant improvements, which indicates that there is not overfitting in the models for the data in our dataset. On another note, L-BFGS does not appear to provide better results than L2SGD. In fact, L2SGD with coefficients for $L2$ regularization seems to provide better values than L-BFGS with similar coefficients for $L2$ regardless of the coefficients for $L1$ regularization.

Among all the configurations, the classifier that better performed with a 3-tokens sliding window was the one that uses L2SGD computing all possible states and transitions, with c2=0.02 as the coefficient for the $L2$ regularization. This configuration obtained a 0.676 as $f1$-score (see Table 3, L2SGD configuration 04). For its part, the classifier that better performed with a 5-tokens sliding window was the one that uses L-BFGS computing all possible states and transitions, with c1=0.0 and c2=0.01 as the coefficient for the $L1$ and $L2$ regularizations, and using the More and Thuente's line search method. This configuration obtained a 0.691 as $f1$-score (see Table 4, L-BFGS configuration 05).

To analyze the performance metrics for these classifiers, we split the dataset in 66% for training and 33% for validation. The metrics per class of sequence, as well as their micro and macro averages, are shown in Table 5. As we can see, the differences between the micro and macro average are very small, which can indicate that those classes of sequence less populated are as well classified as those most populated. Also, the Table shows the values obtained for the same splits using the baseline (see section 4.5). Comparing $f1$-score values for the CRFs approach and the baseline, we can see that they are higher for all the classes of sequence in the case of the CRFs approach. Additionally, when we computed the sequence accuracy for this classifier, i.e, the exact match ratio of sequences that are labelled exactly as in the dataset, we obtained 0.570 (for the configuration L2SGD 04 with 3-tokens sliding window) and 0.593 (for the configuration L-BFGS 05 using 5-tokens sliding window) against the 0.287 obtained for the baseline (see Table 8).

	L2SGD 04; 3-tokens window			L-BFGS 05; 5-tokens window			Baseline			
Sequence	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	support
LOC	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	5
ORG	0.702	0.381	0.494	0.701	0.524	0.599	0.742	0.219	0.338	210
MISC	0.964	0.711	0.818	1.000	0.684	0.813	0.692	0.237	0.353	38
PER	0.758	0.835	0.795	0.804	0.801	0.802	0.954	0.323	0.483	322
MED	0.837	0.610	0.706	0.941	0.542	0.688	0.917	0.186	0.310	59
micro avg	0.763	0.650	0.702	0.792	0.672	0.727	0.867	0.268	0.410	634
macro avg	0.753	0.650	0.682	0.788	0.672	0.719	0.857	0.268	0.407	634

Table 5: Results training for the best classifiers and the baseline considering all class of sequence. Precision, recall, f1-score and support with a split of 66% for training and 33% for validation.

Because the *"location"* class has a low presence in the corpus, in Table 5 we can see an expected result. The support for this class of sequence is really low in the dataset and this influences in the performance metrics of this class. In contrast, the other classes of sequence obtained good values, including the *"miscellaneous"* sequence, despite its reduced support.

In an attempt to improve this accuracy, we can consider whether the number of entries to train the classifier can influence the performance. To do so, we try to get advance of a circumstance previously mentioned: journalists use to personify the reporter and they syntactically use the same grammatical structures in sentences. Thus, we check the use of just one class to identify the reporter regardless of whether it is a person, an organization, etc. Then we consider all classes of sequence (PER, ORG, LOC, MISC) from our dataset as only one class of sequence, namely, the *"reporter"* (R). As a consequence, Table 6 shows the results we obtained. In this Table, we can see how this approach does not provide higher average performance. We can corroborate that the improvement concerning the previous multi-class-of-sequence is not much remarkable, and that as previously the models do not overfit.

Moreover, as previously, we computed the performance metrics for the best classifier. This time, the configuration that best scored was the one with L-BFGS as training algorithm, computes all possible states and transitions, use c1=0.2 as the coefficient for $L1$ regularization and c2=0.2 as the coefficient for $L2$ regularization, and the More and Thuente's line search method (L-BFGS, configuration 21). Performing a split of 66% for training and 33% for evaluation, we obtained the metrics shown in Table 7 for the CRFs configuration and the baseline. Obviusly, baseline results are the same as the obtained for the micro-average shown in Table 5. As we can see, CRFs approaches are significantly more accurate. Furthermore, we obtained a sequence accuracy of 0.555 for the CRFs approach compared to 0.287 for the baseline (see 8).

6. Feature influence and ablation study

To see the influence of each feature set on the classifier, Table 9 details an individual comparison of the performance for every set compared to all the feature set applied to the classifier that uses L-BFGS with configuration 01 for 3-tokens sliding window. As we can see in that table, as expected, the POS tag features are those that most influence in the performance because they are in charge of indicating whether the word is a noun, the main verb, an adverb, etc. Taking into account only the first two characters of the POS tag has a good influence, but the information provided by the whole tag improves the results. The next interesting feature set is suffixes. This can be since the use of specific suffixes in Spanish can designate particular forms and functions of the words, i.e., whether they are acting as an adverb, adjective, noun, etc. The lemma, the role of the word (subject, direct object, etc.) and the related verb are the next feature sets in order of influence. Finally, punctuation and vowel ratios are the features that contribute the least in the process.

To identify the most relevant feature set, we performed an ablation study by systematically removing parts of them following the guidelines provided in [62, 63]. Thus, we started again with all the features

algorithm	cfg	parameters	precision	recall	f1-score	score time	fit time
AROW	01	(AST=False)	0.529 (0.031)	0.530 (0.055)	0.528 (0.038)	5.405 (0.329)	9.848 (0.469)
AROW	02	(AST=True)	0.544 (0.026)	0.548 (0.050)	0.545 (0.029)	5.361 (0.302)	11.173 (0.452)
AROW	03	(AST=True; VAR=0.5)	0.573 (0.031)	0.561 (0.050)	0.565 (0.032)	5.415 (0.336)	11.149 (0.573)
AROW	04	(AST=True; VAR=0.25)	0.592 (0.032)	0.580 (0.056)	**0.585 (0.042)**	5.477 (0.406)	10.829 (0.384)
AP	01	(AST=False)	0.722 (0.035)	0.611 (0.045)	0.662 (0.037)	5.409 (0.328)	10.817 (0.627)
AP	02	(AST=True)	0.713 (0.034)	0.624 (0.041)	**0.665 (0.034)**	5.644 (0.668)	12.661 (0.941)
PA	01	(AST=False; PA type I)	0.718 (0.029)	0.627 (0.048)	0.669 (0.035)	5.322 (0.403)	10.825 (0.196)
PA	02	(AST=True; PA type I)	0.723 (0.037)	0.642 (0.038)	**0.679 (0.033)**	5.344 (0.240)	12.308 (0.440)
PA	03	(AST=False; PA without slack variables)	0.718 (0.029)	0.627 (0.048)	0.669 (0.035)	5.377 (0.288)	11.307 (0.607)
PA	04	(AST=True; PA without slack variables)	0.723 (0.037)	0.642 (0.038)	**0.679 (0.033)**	5.375 (0.287)	12.601 (0.542)
PA	05	(AST=False; PA type II)	0.718 (0.024)	0.625 (0.037)	0.668 (0.027)	5.467 (0.198)	11.819 (0.358)
PA	06	(AST=True; PA type II)	0.715 (0.029)	0.640 (0.041)	0.675 (0.032)	5.322 (0.224)	12.552 (0.508)
PA	07	(AST=False; PA type I; c=0.5)	0.718 (0.029)	0.627 (0.048)	0.669 (0.035)	5.248 (0.214)	10.930 (0.373)
PA	08	(AST=True; PA type I; c=0.5)	0.723 (0.037)	0.642 (0.038)	**0.679 (0.033)**	5.348 (0.397)	12.834 (0.726)
PA	09	(AST=False; PA type II; c=0.5)	0.717 (0.029)	0.626 (0.043)	0.667 (0.031)	5.355 (0.286)	11.330 (0.553)
PA	10	(AST=True; PA type II; c=0.5)	0.713 (0.033)	0.633 (0.037)	0.670 (0.030)	5.166 (0.233)	12.233 (0.493)
L2SGD	01	(AST=False; c2=1.0)	0.772 (0.031)	0.599 (0.045)	0.674 (0.037)	6.009 (1.027)	21.180 (3.875)
L2SGD	02	(AST=True; c2=1.0)	0.772 (0.026)	0.603 (0.047)	0.676 (0.038)	6.228 (1.378)	23.583 (4.869)
L2SGD	03	(AST=True; c2=0.01)	0.764 (0.083)	0.597 (0.044)	0.665 (0.019)	5.396 (0.320)	18.206 (4.013)
L2SGD	04	(AST=True; c2=0.02)	0.764 (0.083)	0.597 (0.045)	0.666 (0.019)	5.381 (0.365)	17.635 (3.476)
L2SGD	05	(AST=True; c2=0.05)	0.765 (0.085)	0.596 (0.043)	0.665 (0.019)	5.643 (0.647)	17.911 (3.436)
L2SGD	06	(AST=True; c2=0.1)	0.768 (0.084)	0.596 (0.042)	0.667 (0.018)	5.388 (0.333)	17.666 (3.462)
L2SGD	07	(AST=True; c2=0.2)	0.734 (0.036)	0.641 (0.036)	**0.684 (0.035)**	5.442 (0.397)	17.903 (3.784)
L-BFGS	01	(AST=False; c1=0.0; c2=1.0)	0.775 (0.032)	0.596 (0.043)	0.673 (0.036)	5.893 (1.039)	44.924 (4.475)
L-BFGS	02	(AST=True; c1=0.0; c2=1.0; LS=MT)	0.779 (0.035)	0.604 (0.040)	0.680 (0.036)	6.561 (1.509)	53.338 (5.859)
L-BFGS	03	(AST=True; c1=0.0; c2=1.0; LS=BT)	0.779 (0.035)	0.604 (0.040)	0.680 (0.036)	6.442 (1.672)	61.390 (9.813)
L-BFGS	04	(AST=True; c1=0.0; c2=1.0; LS=SBT)	0.778 (0.035)	0.604 (0.041)	0.680 (0.036)	6.376 (1.582)	65.619 (10.728)
L-BFGS	05	(AST=True; c1=0.0; c2=0.01; LS=MT)	0.705 (0.030)	0.605 (0.048)	0.650 (0.038)	5.399 (0.276)	157.609 (19.290)
L-BFGS	06	(AST=True; c1=0.0; c2=0.02; LS=MT)	0.722 (0.032)	0.610 (0.042)	0.661 (0.035)	6.402 (1.613)	139.714 (7.752)
L-BFGS	07	(AST=True; c1=0.0; c2=0.05; LS=MT)	0.737 (0.024)	0.611 (0.041)	0.668 (0.032)	5.476 (0.335)	109.543 (8.904)
L-BFGS	08	(AST=True; c1=0.0; c2=0.1; LS=MT)	0.745 (0.028)	0.610 (0.040)	0.670 (0.031)	6.388 (1.524)	101.784 (17.915)
L-BFGS	09	(AST=True; c1=0.0; c2=0.1; LS=BT)	0.744 (0.028)	0.610 (0.039)	0.670 (0.031)	5.446 (0.306)	103.273 (12.860)
L-BFGS	10	(AST=True; c1=0.0; c2=0.2; LS=MT)	0.756 (0.025)	0.613 (0.042)	0.676 (0.032)	5.483 (0.348)	77.024 (6.787)
L-BFGS	11	(AST=True; c1=0.0; c2=0.3; LS=MT)	0.764 (0.028)	0.619 (0.043)	0.684 (0.035)	6.452 (1.599)	76.168 (10.101)
L-BFGS	12	(AST=True; c1=0.0; c2=0.01; LS=MT)	0.779 (0.034)	0.606 (0.038)	0.681 (0.034)	5.301 (0.227)	650.263 (112.381)
L-BFGS	13	(AST=True; c1=0.0; c2=0.02; LS=MT)	0.778 (0.036)	0.605 (0.038)	0.681 (0.035)	5.339 (0.234)	547.758 (118.084)
L-BFGS	14	(AST=True; c1=0.0; c2=0.05; LS=MT)	0.774 (0.036)	0.604 (0.036)	0.678 (0.034)	5.363 (0.367)	649.253 (65.707)
L-BFGS	15	(AST=True; c1=0.1; c2=1.0; LS=MT)	0.773 (0.034)	0.601 (0.036)	0.676 (0.034)	5.259 (0.183)	557.317 (101.066)
L-BFGS	16	(AST=True; c1=0.2; c2=1.0; LS=MT)	0.776 (0.034)	0.599 (0.033)	0.676 (0.032)	6.164 (1.459)	634.993 (55.740)
L-BFGS	17	(AST=True; c1=0.3; c2=1.0; LS=MT)	0.776 (0.033)	0.598 (0.034)	0.675 (0.032)	5.279 (0.300)	666.177 (96.512)
L-BFGS	18	(AST=True; c1=0.1, c2=0.1; LS=MT)	0.750 (0.022)	0.621 (0.036)	0.679 (0.030)	5.237 (0.259)	747.124 (85.681)
L-BFGS	19	(AST=True; c1=0.1, c2=0.2; LS=MT)	0.759 (0.023)	0.622 (0.035)	0.684 (0.030)	5.177 (0.193)	753.683 (43.941)
L-BFGS	20	(AST=True; c1=0.2, c2=0.1; LS=MT)	0.756 (0.033)	0.621 (0.037)	0.682 (0.035)	5.244 (0.269)	754.520 (171.536)
L-BFGS	21	(AST=True; c1=0.2, c2=0.2; LS=MT)	0.765 (0.028)	0.623 (0.033)	**0.686 (0.030)**	5.174 (0.214)	670.808 (157.613)

Table 6: Crossvalidation results training classifiers considering only "reporter" sequence using a 3-tokens sliding window. Mean and standard deviation in parentheses for the precision, recall, f1-score, score time and fit time for each CRFs configuration. AST=all possible states and transitions; c1=coeficient for $L1$ regularization; c2=coeficient for $L2$ regularization; LS=linesearch method (MT=More and Thuente, BT=Backtracking, SBT=StrongBacktracking); c=aggressiveness parameter used for PA-I and PA-II (controls the influence of the slack term on the objective function); VAR=variance

Classifier	precision	recall	f1-score	support
L-BFGS 21 (R)	0.757	0.659	0.705	634
Baseline	0.867	0.268	0.410	634

Table 7: Results training for the best classifier and the baseline considering only "reporter" sequence. Precision, recall, f1-score and support with a split of 66% for training, 33% for validation.

Clasifier	Sequence accuracy
Baseline	0.287
L2SGD 04 (3-tokens sliding window)	0.570
L-BFGS 05 (5-tokens sliding window)	0.593
L-BFGS 21 (R) (3-tokens sliding window)	0.555

Table 8: Sequence accuracy computed for the selected classifiers

Features	precision	recall	f1-score	score time	fit time
All features	0.776 (0.035)	0.571 (0.040)	0.657 (0.031)	6.751 (0.528)	155.986 (16.464)
Word-case	0.742 (0.034)	0.368 (0.046)	0.490 (0.045)	1.997 (0.057)	43.887 (4.474)
Lemma	0.787 (0.051)	0.317 (0.039)	0.449 (0.038)	1.522 (0.346)	16.632 (1.775)
Suffixes	0.799 (0.037)	0.379 (0.033)	0.513 (0.030)	1.702 (0.156)	27.186 (2.924)
Punctuation ratio	0.785 (0.046)	0.256 (0.030)	0.384 (0.034)	1.746 (0.311)	36.273 (0.714)
Vowels ratio	0.788 (0.047)	0.256 (0.032)	0.384 (0.037)	1.268 (0.042)	28.752 (2.886)
POS-tag	0.764 (0.038)	0.451 (0.040)	0.565 (0.034)	1.346 (0.122)	18.234 (1.849)
1st 2-chars POS-tag	0.777 (0.050)	0.399 (0.023)	0.527 (0.028)	1.245 (0.036)	23.714 (2.104)
Role of the word	0.817 (0.041)	0.291 (0.038)	0.427 (0.040)	1.498 (0.358)	27.123 (3.173)
Related verb	0.783 (0.044)	0.289 (0.047)	0.419 (0.049)	1.655 (0.246)	22.747 (1.302)

Table 9: Results for the study on the indivitual comparation of feature sets. The reference configuration is L-BFGS configuration 01 for 3-tokens sliding window.

applied to the classifier that use L-BFGS with configuration 01 for 3-tokens sliding window. Then, we removed the least important feature in each iteration, i.e., the one that caused the smallest decrease in $f1$-score. We repeated these steps until no feature set was left. The reasoning behind this algorithm is that the greatest decrease in performance when removed, the most relevant the feature is, and that feature should be retained. Similarly, the lowest decrease in performance when removed, the least relevant for the classification, and in this case the feature can be removed [62, 63]. Similarly to [62], in the case of a tie, the feature to remove is the one whose individual influence on $f1$-score is lower. We can see the steps of the process in Algorithm 1.

As Table 10 shows, the vowel ratio and the punctuation ratio are the first candidates to be suppressed

Algorithm 1: Procedure for the ablation study.

Result: The influence of each feature set on the classifier.

```
{all} ← all the features;
{remaining} ← {all};
{to_remove} ← ∅;
while {remaining} ≠ ∅ do
    least_relevant ← None;
    lowest_f1_score ← 0;
    foreach f ∈ {remaining} do
        f1_score ← crossvalidateCRF({all} − {to_remove} − {f});
        if f1_score < lowest_f1_score then
            least_relevant ← f;
            lowest_f1_score ← f1_scrore;
        else if f1_score = lowest_f1_score then
            if individual_influence(f) < individual_influence(least_relevant) then
                least_relevant ← f;
            end
    end
    {to_remove} ← {to_remove} ∪ {least_relevant};
    {remaining} ← {remaining} − {least_relevant};
end
```

in this iterative ablation process. This means that these features seem to have the least impact on the classifier. Then, the first two characters of the POStag is the feature that is a candidate for removal. This can be reasonable since its information is supplemented in the entire POStag feature. After that, the role of the word in the sentence (whether it is subject, direct object, etc.) is the next least important feature in the classification process. The related verb, the lemma, and the word case are the three least outstanding features in that order. Finally, as expected, the suffixes and the POStag are the feature sets that contribute most to the performance of the classification process.

The last row of Table 10 show the performance of the classifier only taking into account the word, with no additional feature set.

Features	precision	recall	f1-score	score time	fit time
All features	0.776 (0.035)	0.571 (0.040)	0.657 (0.031)	6.751 (0.528)	155.986 (16.464)
–Word-case (W)	0.771 (0.033)	0.531 (0.041)	0.628 (0.032)	4.645 (0.405)	70.249 (6.374)
–Lemma (L)	0.776 (0.033)	0.560 (0.042)	0.650 (0.033)	6.680 (1.571)	94.634 (7.584)
–Suffixes (S)	0.783 (0.031)	0.522 (0.053)	0.625 (0.042)	4.875 (0.329)	91.341 (11.693)
–Punctuation ratio (PR)	0.774 (0.034)	0.559 (0.045)	0.648 (0.037)	5.930 (1.378)	86.281 (15.218)
–Vowels ratio (V)	0.776 (0.033)	0.561 (0.044)	**0.650 (0.034)**	5.209 (0.279)	81.160 (7.208)
–POS-tag (P)	0.782 (0.030)	0.555 (0.044)	0.648 (0.033)	5.045 (0.178)	85.974 (9.676)
–1st 2-chars POS-tag (2P)	0.775 (0.030)	0.554 (0.042)	0.645 (0.032)	6.219 (1.276)	78.606 (5.426)
–Role of the word (R)	0.774 (0.033)	0.556 (0.043)	0.646 (0.032)	5.999 (1.510)	88.623 (8.753)
–Related verb (RV)	0.765 (0.033)	0.554 (0.026)	0.642 (0.023)	6.053 (1.383)	94.218 (9.906)
–V–W	0.770 (0.034)	0.530 (0.042)	0.627 (0.033)	4.899 (0.871)	56.737 (5.312)
–V–L	0.775 (0.036)	0.553 (0.044)	0.645 (0.036)	6.012 (1.032)	87.893 (8.392)
–V–S	0.783 (0.033)	0.523 (0.052)	0.625 (0.040)	4.519 (0.281)	78.867 (9.824)
–V–PR	0.774 (0.033)	0.560 (0.046)	**0.649 (0.037)**	5.619 (1.255)	67.164 (7.128)
–V–P	0.781 (0.030)	0.554 (0.043)	0.647 (0.032)	6.594 (0.913)	99.113 (9.736)
–V–2P	0.776 (0.030)	0.554 (0.041)	0.646 (0.032)	5.706 (1.478)	81.781 (10.660)
–V–R	0.774 (0.033)	0.556 (0.044)	0.645 (0.032)	4.858 (0.318)	73.253 (7.671)
–V–RV	0.767 (0.033)	0.553 (0.026)	0.642 (0.022)	5.141 (0.547)	85.024 (10.082)
–V–PR–W	0.770 (0.035)	0.531 (0.041)	0.628 (0.032)	4.386 (0.940)	44.750 (4.899)
–V–PR–L	0.773 (0.037)	0.552 (0.044)	0.643 (0.037)	5.324 (1.194)	77.690 (6.696)
–V–PR–S	0.784 (0.033)	0.524 (0.052)	0.626 (0.040)	5.614 (0.948)	78.083 (11.000)
–V–PR–P	0.779 (0.031)	0.552 (0.045)	0.645 (0.035)	5.436 (1.220)	77.281 (8.751)
–V–PR–2P	0.776 (0.031)	0.555 (0.042)	**0.646 (0.033)**	5.415 (1.075)	68.752 (8.639)
–V–PR–R	0.773 (0.031)	0.555 (0.046)	0.645 (0.033)	4.662 (0.344)	68.085 (10.420)
–V–PR–RV	0.766 (0.031)	0.552 (0.028)	0.641 (0.024)	5.261 (1.184)	67.233 (6.056)
–V–PR–2P–W	0.773 (0.033)	0.524 (0.044)	0.623 (0.033)	3.162 (0.108)	36.337 (1.450)
–V–PR–2P–L	0.776 (0.037)	0.547 (0.046)	0.641 (0.038)	4.680 (1.076)	60.106 (8.272)
–V–PR–2P–S	0.786 (0.034)	0.517 (0.052)	0.622 (0.041)	3.816 (0.244)	60.073 (5.413)
–V–PR–2P–P	0.779 (0.028)	0.522 (0.061)	0.623 (0.048)	4.239 (0.292)	57.628 (2.381)
–V–PR–2P–R	0.777 (0.031)	0.549 (0.049)	**0.641 (0.035)**	4.748 (1.084)	53.501 (3.145)
–V–PR–2P–RV	0.769 (0.033)	0.548 (0.026)	0.639 (0.023)	4.908 (0.976)	62.312 (8.205)
–V–PR–2P–R–W	0.770 (0.031)	0.520 (0.042)	0.619 (0.032)	2.801 (0.116)	26.220 (2.376)
–V–PR–2P–R–L	0.773 (0.029)	0.542 (0.051)	0.635 (0.039)	4.251 (0.942)	55.359 (5.128)
–V–PR–2P–R–S	0.781 (0.033)	0.512 (0.054)	0.616 (0.041)	3.311 (0.255)	46.899 (2.952)
–V–PR–2P–R–P	0.767 (0.038)	0.512 (0.059)	0.612 (0.046)	6.102 (2.081)	69.618 (10.250)
–V–PR–2P–R–RV	0.768 (0.028)	0.544 (0.033)	**0.636 (0.027)**	3.853 (0.273)	53.407 (5.065)
–V–PR–2P–R–RV–W	0.777 (0.033)	0.524 (0.032)	0.625 (0.025)	2.635 (0.265)	25.020 (1.699)
–V–PR–2P–R–RV–L	0.769 (0.035)	0.541 (0.031)	**0.634 (0.027)**	3.529 (0.678)	52.693 (5.820)
–V–PR–2P–R–RV–S	0.774 (0.037)	0.497 (0.048)	0.603 (0.039)	3.335 (0.746)	45.932 (4.330)
–V–PR–2P–R–RV–P	0.766 (0.030)	0.498 (0.045)	0.602 (0.037)	3.369 (0.259)	48.373 (5.266)
–V–PR–2P–R–RV–L–W	0.775 (0.040)	0.510 (0.032)	**0.614 (0.030)**	2.143 (0.101)	23.488 (2.012)
–V–PR–2P–R–RV–L–S	0.771 (0.040)	0.468 (0.043)	0.581 (0.037)	2.383 (0.059)	45.478 (5.091)
–V–PR–2P–R–RV–L–P	0.761 (0.028)	0.486 (0.048)	0.592 (0.038)	2.673 (0.072)	47.971 (3.362)
–V–PR–2P–R–RV–L–W–S	0.764 (0.038)	0.451 (0.040)	**0.565 (0.034)**	1.329 (0.109)	16.614 (1.212)
–V–PR–2P–R–RV–L–W–P	0.799 (0.037)	0.379 (0.033)	0.513 (0.030)	2.002 (0.560)	23.795 (0.967)
–V–PR–2P–R–RV–L–W–S–P	0.779 (0.046)	0.254 (0.036)	0.381 (0.042)	0.907 (0.051)	19.269 (2.880)

Table 10: Results for the ablation study. The reference configuration is L-BFGS configuration 01 for 1-tokens sliding window. The character '–' means substraction. Numbers in bold indicate the $f1$-score of the candidate feature sets to remove.

7. Conclusion

With the aim to provide tools that help on building automatic systems to support the journalistic transparency against fake news, in this article we have proposed to automatically extract the sources of information in newspaper articles so that their veracity can be verified. To achieve this, we make use of Natural Language Processing (NLP) and Machine Learning (ML) to automate the extraction of that relevant information.

Consequently, we have detailed the application of Conditional Random Fields (CRFs) to recognize a specific type of entity we have called the *"reporter"* in newspaper articles for the Spanish language. Thus, we have carried out an experimental setup in which different CRFs configurations have been defined, validated and analyzed to identify the best of them, to compare it against a defined baseline. Furthermore, we have examined the influence of the different feature sets in the classification performance, and also, defined and performed an ablation process systematically to identify the most relevant feature set.

As a consequence, we have obtained the initial results and baseline for our goal, and also, we have created a labelled corpus that other researchers can use and improve. Thus, this article contributes to the state of art in the application of CRFs for a specific type of Entity Recognition task.

Improving the performance of the approach by introducing new and/or different features and configurations, comparing the results with other approximations such as those implemented through Neural Networks, extending the dataset, etc. are some of the tasks we have established as future work.

Acknowledgments

This research work has been co-funded by Display Connectors S.L. through the project entitled "Identifying relevant entities in newspaper articles"(in Spanish *"Identificación de entidades relevantes en noticias periodísticas"*), and by the Madrid Regional Government through the project e-Madrid-CM (P2018/TCS-4307). The e-Madrid-CM project is also co-financed by the Structural Funds (FSE and FEDER). Also, we give special thanks to the people from the Público online newspaper for their work and support.

References

[1] B. McNair, Fake News: Falsehood, Fabrication and Fantasy in Journalism, Disruptions: studies in digital journalism, Routledge, London, 2017. `doi:10.4324/9781315142036`.

[2] M. Karlsson, The immediacy of online news, the visibility of journalistic processes and a restructuring of journalistic authority, Journalism (2011). `doi:10.1177/1464884910388223`.

[3] M. Revers, The twitterization of news making: Transparency and journalistic professionalism, Journal of Communication (2014). `doi:10.1111/jcom.12111`.

[4] K. Chadha, M. Koliska, Newsrooms and transparency in the digital age, Journalism Practice (2015). `doi:10.1080/17512786.2014.924737`.

[5] T. P. Vos, S. Craft, The discursive construction of journalistic transparency, Journalism Studies (2017). doi:10.1080/1461670X.2015.1135754.

[6] D. Nadeau, A survey of named entity recognition and classification, Linguisticae Investigationes (30) (2007) 3–26. doi:10.1075/li.30.1.03nad.

[7] S. K. Das, S. Dhar, Entity recognition in bengali language, in: Proceedings of the International Symposium on Advanced Computing and Communication, ISACC'2016, 2016, pp. 157–160. doi:10.1109/ISACC.2015.7377333.

[8] N. Mahanta, S. Dhar, S. Roy, Entity Recognition in Assamese Text, in: Proceedings of the International Conference on Communication and Electronics Systems, ICCES'2016, IEEE, 2016, pp. 1–5. doi:10.1109/CESYS.2016.7890006.

[9] F. Alam, B. Magnini, R. Zanoli, Comparing named entity recognition on transcriptions and written texts, Studies in Computational Intelligence (2015). doi:10.1007/978-3-319-14206-7_4.

[10] I. Yamada, T. Ito, H. Takeda, Y. Takefuji, Linkify: Enhancing Text Reading Experience by Detecting and Linking Helpful Entities to Users (2018). doi:10.1109/MIS.2018.111144233.

[11] S. Malhotra, A. Dixit, Article: An effective approach for news article summarization, International Journal of Computer Applications 76 (16) (2013) 5–10.

[12] C. Barros, E. Lloret, E. Saquete, B. Navarro-Colorado, Natsum: Narrative abstractive summarization through cross-document timeline generation, Information Processing & Management 56 (5) (2019) 1775 – 1793. doi:https://doi.org/10.1016/j.ipm.2019.02.010.
URL http://www.sciencedirect.com/science/article/pii/S0306457318305922

[13] T. Takada, M. Arai, T. Takagi, Automatic keyword annotation system using newspapers, Journal of Advanced Computational Intelligence and Intelligent Informatics (2014). doi:10.1109/SCIS-ISIS.2012.6505157.

[14] P. C. Cardoso, T. A. Pardo, M. Taboada, Subtopic annotation and automatic segmentation for news texts in Brazilian Portuguese, Corpora (2017). doi:10.3366/cor.2017.0108.

[15] V. L. Rubin, Y. Chen, N. J. Conroy, Deception Detection for News: Three Types of Fake News, Vol. 52 of ASIST'2015, 2015, pp. 1–4. doi:10.1002/pra2.2015.145052010083.

[16] R. Stecanella, J. Bonanata, D. Wonsever, A. Rosá, Opinion Search in Spanish Written Press, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (2014). doi:10.1007/978-3-319-12027-0_25.

[17] K. Ralf, B. Sabine, W. René, Modeling Human Newspaper Readers: The Fuzzy Believer approach, Natural Language Engineering (2014). doi:10.1017/S1351324912000289.

[18] D. Najar, S. Mesfar, Opinion Mining and Sentiment Analysis for Arabic On-line Texts: Application on the Political Domain, International Journal of Speech Technology (2017). doi:10.1007/s10772-017-9422-4.

[19] H. Rahab, A. Zitouni, M. Djoudi, SIAAC: Sentiment Polarity Identification on Arabic Algerian Newspaper Comments, Advances in Intelligent Systems and Computing (2018). doi:10.1007/978-3-319-67621-0_12.

[20] I. Afolabi, O. Sowunmi, O. Daramola, Semantic Association Rule Mining in Text Using Domain Ontology (2017). doi:10.1504/IJMSO.2017.087646.

[21] A. Pinto, H. Gonçalo Oliveira, Á. Figueira, A. O. Alves, Predicting the Relevance of Social Media Posts Based on Linguistic Features and Journalistic Criteria, New Generation Computing (2017). doi:10.1007/s00354-017-0015-1.

[22] L. Gatti, G. Ozbal, M. Guerini, O. Stock, C. Strapparava, Heady-lines: A creative generator of newspaper headlines, in: Companion Publication of the 21st International Conference on Intelligent User Interfaces, IUI '16 Companion, ACM, New York, NY, USA, 2016, pp. 79–83. doi:10.1145/2876456.2879469.
URL http://doi.acm.org/10.1145/2876456.2879469

[23] X. Ao, P. Luo, C. Li, F. Zhuang, Q. He, Discovering and learning sensational episodes of news events, Information Systems 78 (2018) 68 – 80. doi:10.1016/j.is.2018.05.003.

URL http://www.sciencedirect.com/science/article/pii/S0306437916303520

[24] M. Guerini, C. Strapparava, Why do urban legends go viral?, Information Processing & Management 52 (1) (2016) 163 – 172, emotion and Sentiment in Social and Expressive Media. doi:https://doi.org/10.1016/j.ipm.2015.05.003. URL http://www.sciencedirect.com/science/article/pii/S0306457315000540

[25] T. O'Keefe, S. Pareti, J. Curran, I. Koprinska, M. Honnibal, A sequence labelling approach to quote attribution, in: Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Proceedings of the Conference, EMNLP-CoNLL 2012, 2012, pp. 790–799.

[26] R. Prasad, N. Dinesh, A. Lee, A. Joshi, B. Webber, Annotating attribution in the penn discourse treebank, in: Proceedings of the Workshop on Sentiment and Subjectivity in Text, SST '06, Association for Computational Linguistics, Stroudsburg, PA, USA, 2006, pp. 31–38.
URL http://dl.acm.org/citation.cfm?id=1654641.1654646

[27] S. Pareti, T. O'Keefe, I. Konstas, J. R. Curran, I. Koprinska, Automatically detecting and attributing indirect quotations, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Seattle, Washington, USA, 2013, pp. 989–999.
URL https://www.aclweb.org/anthology/D13-1101

[28] M. S. C. Almeida, M. B. Almeida, A. F. T. Martins, A joint model for quotation attribution and coreference resolution, in: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Gothenburg, Sweden, 2014, pp. 39–48. doi:10.3115/v1/E14-1005.
URL https://www.aclweb.org/anthology/E14-1005

[29] H. Zhang, F. Boons, R. Batista-Navarro, Whose story is it anyway? automatic extraction of accounts from news articles, Information Processing & Management 56 (5) (2019) 1837 – 1848. doi:https://doi.org/10.1016/j.ipm.2019.02.012.
URL http://www.sciencedirect.com/science/article/pii/S0306457318306101

[30] J. A. Bilmes, What hmms can do, IEICE - Trans. Inf. Syst. E89-D (3) (2006) 869–891. doi:10.1093/ietisy/e89-d.3.869.

[31] J. D. Lafferty, A. McCallum, F. C. N. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Proceedings of the Eighteenth International Conference on Machine Learning, ICML'2001, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001, pp. 282–289.

[32] C. Sutton, A. McCallum, An Introduction to Conditional Random Fields for Relational Learning, Graphical Models 7 (2002) 93. doi:10.1677/JME-08-0087.

[33] C. Sutton, A. McCallum, An Introduction to Conditional Random Fields, Machine Learning (2010). doi:10.1561/2200000013.

[34] C. Sutton, A. McCallum, An introduction to conditional random fields, Found. Trends Mach. Learn. 4 (4) (2012) 267–373. doi:10.1561/2200000013.

[35] G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. Hakkani-Tur, X. He, L. Heck, G. Tur, D. Yu, G. Zweig, Using recurrent neural networks for slot filling in spoken language understanding, IEEE/ACM Transactions on Audio, Speech, and Language Processing 23 (3) (2015) 530–539. doi:10.1109/TASLP.2014.2383614.

[36] Z. Huang, W. Xu, K. Yu, Bidirectional lstm-crf models for sequence tagging, cite arxiv:1508.01991 (2015).
URL http://arxiv.org/abs/1508.01991

[37] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, Computing Research Repository abs/1603.01360 (2016). arXiv:1603.01360.
URL http://arxiv.org/abs/1603.01360

[38] J. P. Chiu, E. Nichols, Named entity recognition with bidirectional LSTM-CNNs, Transactions of the Association for Computational Linguistics 4 (2016) 357–370. doi:10.1162/tacl_a_00104.
URL https://www.aclweb.org/anthology/Q16-1026

[39] F. Zhai, S. Potdar, B. Xiang, B. Zhou, Neural models for sequence chunking, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI'2017, 2017.

[40] Y. Wang, Y. Shen, H. Jin, A bi-model based rnn semantic frame parsing model for intent detection and slot filling, in: Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics, NAACL-HLT'2018, New Orleans, Louisiana, 2018, p. 309–314.

[41] A. Graves, Studies in Computational Intelligence, Springer Berlin Heidelberg, 2015. `doi:10.1007/978-3-642-24797-2`.

[42] a. McCallum, W. Li, Early Results for Named Entity Recognition with Conditional Random Fields, Proceedings of CoNLL-2003 (2003) 188–191`doi:10.3115/1119176.1119206`.

[43] G. Bekoulis, J. Deleu, T. Demeester, C. Develder, Joint entity recognition and relation extraction as a multi-head selection problem, Expert Systems with Applications 114 (2018) 34–45. `doi:10.1016/j.eswa.2018.07.032`.

[44] Y. Choi, C. Cardie, E. Riloff, S. Patwardhan, Identifying sources of opinions with conditional random fields and extraction patterns, in: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT'2005, Association for Computational Linguistics, Stroudsburg, PA, USA, 2005, pp. 355–362. `doi:10.3115/1220575.1220620`.

[45] T. Nakagawa, K. Inui, S. Kurohashi, Dependency Tree-based Sentiment Classification using CRFs with Hidden Variables, Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL (June) (2010) 786–794. `doi:10.3115/1220175.1220274`.

[46] T. Chen, R. Xu, Y. He, X. Wang, Improving sentiment analysis via sentence type classification using bilstm-crf and cnn, Expert Systems with Applications 72 (2017) 221–230. `doi:10.1016/j.eswa.2016.10.065`.

[47] M. Mitchell, J. Aguilar, T. Wilson, B. Van Durme, Open Domain Targeted Sentiment, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, no. October in EMNLP'2013, 2013, pp. 1643–1654.

[48] M. Zhang, Y. Zhang, D. T. Vo, Neural Networks for Open Domain Targeted Sentiment, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, no. September in EMNLP'2015, 2015, pp. 612–621. `doi:10.18653/v1/D15-1073`.

[49] A. Sadeghian, L. Sundaram, D. Z. Wang, W. F. Hamilton, K. Branting, C. Pfeifer, Automatic semantic edge labeling over legal citation graphs, Artificial Intelligence and Law 26 (2) (2018) 127–144. `doi:10.1007/s10506-018-9217-1`.

[50] J. Nocedal, Updating quasi-newton matrices with limited storage, Mathematics of computation 35 (151) (1980) 773–782. `doi:10.1090/S0025-5718-1980-0572855-7`.

[51] S. Shalev-Shwartz, Y. Singer, N. Srebro, A. Cotter, Pegasos: Primal estimated sub-gradient solver for svm, Mathematical Programming 127 (1) (2011) 3–30. `doi:10.1007/s10107-010-0420-4`.
URL `https://doi.org/10.1007/s10107-010-0420-4`

[52] M. Collins, Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing - Volume 10, EMNLP'2002, Association for Computational Linguistics, Stroudsburg, PA, USA, 2002, pp. 1–8. `doi:10.3115/1118693.1118694`.
URL `https://doi.org/10.3115/1118693.1118694`

[53] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, Y. Singer, Online passive-aggressive algorithms, The Journal of Machine Learning Research 7 (2006) 551–585.
URL `http://dl.acm.org/citation.cfm?id=1248547.1248566`

[54] A. Mejer, K. Crammer, Confidence in structured-prediction using confidence-weighted models, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Cambridge, MA, 2010, pp. 971–981.
URL `https://www.aclweb.org/anthology/D10-1095`

[55] T. De Smedt, W. Daelemans, Pattern for python, Journal of Machine Learning Research 13 (2012) 2031–2035.

[56] N. Okazaki, Crfsuite: a fast implementation of conditional random fields (crfs), last accessed on 10/8/2019 (2007).
URL http://www.chokkan.org/software/crfsuite/

[57] N. Okazaki, libLBFGS: a library of Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS), last accessed on 10/8/2019 (2014).
URL http://www.chokkan.org/software/liblbfgs/

[58] T. Peng, M. Korobov, Python-CRFsuite, last accessed on 10/8/2019 (2018).
URL https://python-crfsuite.readthedocs.io/en/latest/

[59] M. Korobov, sklearn-crfsuite, last accessed on 10/8/2019 (2015).
URL https://sklearn-crfsuite.readthedocs.io

[60] Hironsan, seqeval: A python framework for sequence labeling evaluation, last accessed on 10/8/2019 (2019).
URL https://github.com/chakki-works/seqeval

[61] spacy, spacy · industrial-strength natural language processing in python, last accessed on 10/8/2019 (2019).
URL https://spacy.io/

[62] S. J. Bethard, Finding event, temporal and causal structure in text:a machine learning approach, Ph.D. thesis, University of Colorado (2008).

[63] K. C. Fraser, G. Hirst, N. L. Graham, J. A. Meltzer, S. E. Black, E. Rochon, Comparison of different feature sets for identification of variants in progressive aphasia, in: roceedings of the Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality, Association for Computational Linguistics, Baltimore, Maryland, USA, 2014, pp. 17–26. doi:10.3115/v1/W14-3203.
URL https://www.aclweb.org/anthology/W14-3203