NLPyPort: Named Entity Recognition with CRF and Rule-Based Relation Extraction

João Ferreira[1], Hugo Gonçalo Oliveira[1,2][0000−0002−5779−8645], and Ricardo Rodrigues[2,3][0000−0002−6262−7920]

[1] Department of Informatics Engineering of the University of Coimbra, Portugal
[2] Center for Informatics and Systems of the University of Coimbra, Portugal
[3] College of Education of the Polytechnic Institute of Coimbra, Portugal
jdcoelho@student.dei.uc.pt, hroliv@dei.uc.pt, rmanuel@dei.uc.pt

Abstract. This paper describes the application of the NLPyPort pipeline to Named Entity Recognition (NER) and Relation Extraction in Portuguese, more precisely in the scope of the IberLEF-2019 evaluation task on the topic. NER was tackled with CRF, based on several features, and trained in the HAREM collection, but results were low. This was partly caused by an issue on the submitted model, which had been trained in lowercase text, but, apparently, also due to the training data used, which highlights the different natures of HAREM, the source of the majority of the testing corpus, and SIGARRA. Relations were extracted with a set of rules bootstrapped from the examples provided by the organisation. Despite an F1-score of 0.72, we were the only participants in this task. We also express our doubts concerning the utility of the extracted relations.

Keywords: NLP · NER · CRF · Relation Extraction · PoS Tagging · Pattern Based

1 Introduction

Natural Language Processing (NLP) often starts with initial (pre-)processing tasks that add structural and linguistic information to the text. Modules for those initial tasks are often assembled in pipelines, which are thus the cornerstone of NLP. Ensuring the best results for each of those modules will often provide better results for higher-level applications that rely on them.

Despite the existence of NLP pipelines with modules ready to use or trainable for different languages, they generally fail to target language-specific aspects. Having this in mind, we worked on improving some modules of the Natural Language Toolkit (NLTK) [3], a popular NLP pipeline in Python, for Portuguese. This resulted in NLPyPort [8], which tackles similar aspects as its Java counterpart, NLPPort [16].

Initial improvements focused on Tokenization and Part-of-Speech (PoS) Tagging, and a new Lemmatization module was developed. A module for Named Entity Recognition (NER) was then developed, based on Conditional Random Fields (CRF) [11] for predicting the BIO tags of the entities, and assembled into the pipeline using the CRF Suite [13]. More recently, we started to work on a module for fact extraction, currently based on rules composed by PoS tags, discovered in a bootstrapping fashion.

This paper describes how NLPyPort was used in the IberLEF 2019 [6] tasks on Portuguese Named Entity Recognition and Relation Extraction. More precisely, tasks 1 and 2 were addressed, respectively Named Entity Recognition [5] and Relation Extraction for Named Entities [1]. This participation allowed us to test the pipeline, as well as to pin-point possible existing problems.

In the next sections, the architectural design is described as follows: Section 2 briefly describes the CRF configurations used in this shared task. Section 3 describes how the current version of FactPyPort, our fact extraction, was developed, using PoS tags as rules. Section 4 analyses the results obtained for both NER and Relation extraction Tasks, and explains the reasons found for failure with pos-evaluation experiments. Section 6 concludes the paper and discusses lines for future work.

2 CRFs for Named Entity Recognition

In order to identify entities, a new NER module was assembled and implemented into NLPyPort, using as a starting point the already existing CRF Suite. In the past, CRFs have lead to good results in sequence processing, including NER. Moreover, using CRF Suite revealed to be much simpler to train than NLTK's NER module, and at least as easy to use. Although it relies on text in the CoNLL 2003 format, using BIO tags for delimiting entities, this format is quite common and easy to convert to.

Training a new module only requires new data, while features to exploit can also be set accordingly. In our case, we went further than just the sequence of words and exploited the following features, in a 5-token context window: punctuation sign; only ASCII characters; only lowercase characters; only uppercase characters; only alphabetic characters; only numbers; alphanumeric; starts with an uppercase character; ends with an 'a'; ends with with a 's'; token shape ('A' means uppercase, 'a' lowercase, '#' number, '-' punctuation); length; prefixes and suffixes with lengths from 1 to 5.

The PoS tags and lemmatized words were obtained using the previous modules of the pipeline. These features are the same Lopes et al. [12] used for NER in Portuguese clinical text.

As mentioned earlier, our model could be trained with any collection where named entities (NEs) are annotated, using the CoNLL 2003 format. For Portuguese, we know of three collections of this kind: the First and the Second HAREM [17, 9]; and SIGARRA [15]. Though not originally created in this format, both of them were later made available this way, in the scope of

André Pires MSc thesis [15](`https://github.com/arop/ner-re-pt`), though with some simplifications, in the case of HAREM — e.g., removal of the type attribute. However, besides not tackling exactly the same categories, an important difference between the collections of HAREM and SIGARRA is that the former was annotated and extensively revised by a team of five people, following well-defined guidelines, while the latter, to the best of our knowledge, was annotated by a single MSc student. This is mainly why we decided to train our model only on the HAREM collection. To meet the shared task guidelines, we changed the names of the entity categories accordingly.

3 FactPyPort

The FactPyPort module is the most recent addition to the NLPyPort pipeline, with its current version developed specifically for extracting relations in the scope of this shared task. It thus deserves a more detailed description. As it happened in this task, given the delimitation of NEs, FactPyPort relies on a set of examples for generating a number of PoS-based rules in a bootstrapping fashion, inspired by earlier work by Hearst [10] or by Pantel & Pennachiotti [14].

3.1 PoS tag-based patterns

The core idea of the developed system was to take advantage of the existing patterns between words, while maintaining somewhat of an open set of relations. Given that the goal of the shared task was not to find relations of a specific type, but rather words that described the relation, it was decided that all the rules had to rely on patterns valid for any sentence. A way to do this kind of generalisation was to consider not the words themselves, but rather their PoS tags.

This way, the system could take a sentence with previously annotated NEs and the words that described the relations, then save the configurations of PoS tags that correspond to the relation, and use this as a new rule. After extracting a set of rules, the system assumes that a relation is present in a sentence if the PoS-tag sequence matches those of a rule, and outputs the corresponding words.

3.2 Learning, ranking and rules

In order to learn the PoS patterns necessary for identifying relations, a set of annotated example sentences was given. The following is an example of a pattern converted into a rule:

- `1§0111010§punc art adj prp n prp art`

The rule is divided into three parts, denoted by the § symbol. The first part (in this case, a `1`) is the rule ranking. Since the final system ended up being sorted by rule size, this was not used. In previous versions of the system, rules were ranked by frequency, saved here.

The second part is a set of 1s and 0s. There is a number for each of the following tags, for all the words between the two NEs. A 1 indicates that the word with the corresponding PoS tag is part of the relation, while the 0 indicates that it is not a part of the relation.

As mentioned earlier, the last part is the sequence of PoS tags between the two NEs.

To extract a new rule from the annotated data, the system first gets the words between the two NEs and their PoS tags. After this, it finds the position of the words indicated as being part of the relation by setting the number in the corresponding indices to 1.

In order to ensure better results, the rules are sorted by size. This way, we start with higher specificity and low coverage, and progress to a point of lower specificity and higher coverage. A highly specific rule can look somewhat like:

- 1§000000000000000010000100000111§art n punc pron pron v adv art n n conj adv v v adv v prp art n punc v prp art n n punc punc prp art n

While a high coverage rule can be as simple as:

- 1§1§punc

3.3 Generating additional rules

In order to get more rules from one example, a second step is made. It consists of striping the sentence of the tags with 0. The result is a smaller rule with only 1s, which means all the words are part of the relation. This can be useful because, even if a longer sentence is not an exact match, parts of it may be and therefore a relation may be found.

This is illustrated with the following example, with a detected rule and the reduced generated rule. First detected rule:

- 1§000000010000100011§prp prop punc pron n v n adv art n n conj v prp art n n adv

Reduced generated rule:

- 1§1111§adv v n adv

3.4 Training data

One of the challenges that come with training a system of this kind is the low number of available examples that identify the relations, especially those adhering to the shared task guidelines. For this reason, the training data considered comprised only the 100 examples released by the task organisers. We believed, and came also to the conclusion, that the system trained with only these was fairly good, but that the results of the system could be improved if trained with more data. The PoS tags necessary for training were obtained with NLPyPort.

4 Evaluation and Result analysis

In the following sections, we present the obtained results for each of the tasks, discuss them and share thoughts on how they can be improved.

4.1 NER Results

The results for the Task 1, in Table Table 1, took us by surprise, since we did not expect them to be so low.

Corpus	**Category**	**Precision**	**Recall**	**F1**
Police Dataset	PER	21.72%	53.07%	30.83
Clinical Dataset	PER	27.27%	26.25%	26.75
General Dataset	Overall	26.08%	19.78%	22.50
	ORG	19.41%	12.62%	15.30
	PER	50.07%	34.34%	40.74
	PLC	42.31%	20.64%	27.75
	TME	8.99%	11.11%	9.94
	VAL	56.60%	54.55%	55.56

Table 1. Results for the NER task, by entity category, with a CRF trained on the HAREM collection (official).

Once we saw the results, we noticed that we had trained our model in the same corpus where some sentences were taken from, HAREM. In fact, the higher performance for the VAL category is explained by this fact. But it made us think why it was not even higher.

To come up with a reason, we started inspecting the process that lead to the creation of the model, and soon noticed that we had submitted a faulty model, trained in a lowercase version of the corpus, where an important feature of entities of many categories is lost (starting with uppercase). We cannot say much about the results in the clinical and police datasets, because they are not available. Yet, after fixing the previous issue and training in the same collection, for the general dataset, our results were improved. The main difference was that NEs of the VAL category were now perfectly recognised, as expected, given that the training data was the same as the testing. Also because of this, it should be perceived as a meaningless result. Moreover, F1 improved about 5 points for all the remaining categories, except TME, for which the tagging in HAREM is probably inconsistent with SIGARRA. Even with this improvement, performance is quite low, which suggests that SIGARRA and HAREM are very different, and a (sequence prediction) model trained in one will not perform well on the other.

Moving on, we could not help thinking what would have happened if, instead of training the model with HAREM, we had opted instead to train it with SIGARRA, or even with both HAREM and SIGARRA, given that they are both publicly available and ready to use by our CRF. Therefore, in order to validate the model, we trained the CRF with both SIGARRA and HAREM and tested it in the general dataset.

As expected, F1 was above 92 for all categories, with an overall result of 95.36. Although this helped us confirm that the model is working well, we stress that it is by no means an indicator of its quality, because all testing data was included in the training collections.

Even though the results were unsatisfying, our participation in this task helped us validating and fixing some issues with our NER module. Plus, we showed that training the CRF with the HAREM corpus is not enough to achieve a good performance in NER for Portuguese in any textual source. Anyway, with more data, it should be possible to improve the results obtained.

4.2 Relation Extraction Results

The rule-based version of FactPyPort was used for Task 2 — Relation Extraction for Named Entities. In this task, NEs were already tagged, thus making the goal of the system easier: to identify relations between the tagged NEs. FactPyPort was trained according to the description in Section 3 and the results obtained this way are in Table 2. The "Exactly" column measures when the system was able to select all the words that are part of the relation, and "Partial" considers also cases when only part of the relation was identified.

Exactly			**Partial**		
Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
0.736	0.711	0.7235	0.7662	0.748	0.757

Table 2. Results of the FactPyPort system.

Despite the simplicity of the model, results in this task were interesting. However, being the only system participating in this task, we do not have a baseline to which we can compare against.

Still, upon analysing the results obtained and comparing them with the expected results, we may speculate that a possible explanation for the high results, given the reduced training data, could be the test data itself. For the three most frequent relations in the testing data, Table 3 shows their number of occurrences and the corresponding proportion. Those were also frequent relations in the examples provided by the organization and used for bootstrapping the rules. All relations of those three types were correctly identified (partially, in a minority of situations), with the system additionally suggesting some additional (incorrect) relations of this kind, especially of the type *em*. Table 4 has examples of the latter. This suggests that there is some over-fitting.

We further noticed that five identified relations had an empty type ("") (see Table 4), even though, in the expected output, there were no instances of this

	Relation	**Occurrences**
Test data	de	43 (28.8%)
	(	18 (12.1%)
	em	13 (8.7%)
Results	de	45 (30%)
	(	19 (12.8%)
	em	20 (13.4%)

Table 3. Top most frequent relations in the testing data and gotten results.

Sentence	**Expected**	**Result**
...de Nova_Iorque, Rudy_Giuliani, ... foi presidente de a Câmara ...	presidente de	de
...registo, Martin_SCHULZ (PSE, DE), líder de o PSE, recordou que, em a...	lider de	(
...Helena_Roseta ...em as próximas eleições intercalares para a Câmara_de_Lisboa.	eleições para	em
... redor de Sarajevo 'como último recurso, para defender as tropas de a ONU' ...	defender	

Table 4. Example cases where there extra cases of the most commonly relations were wrongly classified (top three) or where no relation was found (bottom one)

kind. This happens because the system tries to find a relation between the given NEs and, when no rule matches, it returns an empty relation. This could be turned off, and possibly minimised with more training data.

5 Working notes

The NLPyPort pipeline and its source code is available at `https://github.com/jdportugal/NLPyPort`. The source code for running the NER system used for Task 1, for both the submitted and the current versions, is available at `https://github.com/jdportugal/CRFIBERLEF`. The README file contains all the instructions.

The FactPyPort system used for Task 2 is available at `https://github.com/jdportugal/FactPyPortIBERLEF`. For running it, all the requirements should be installed (`requirements.txt` file) and the following line should be ran on the terminal:

```
≫ python FactPort.py input_file > output_file
```

The output file will contain the results in the format specified for Task 2.

6 Conclusions and Future Work

Named Entity Recognition and Relation Extraction are arguably highly-relevant NLP tasks. As time goes by, NER for the Portuguese language is improving and getting results closer to those for English, but, at the same time, there is still much room for improvement. In fact, we ended up confirming that training a CRF in one of the most popular collections with annotated NEs in Portuguese is not enough for a good-performing model in another collection where NEs of similar categories are annotated. Coming up with a proper reason for this would require further analysis, but the different nature of both collections, as well as their annotation guidelines and processing might have an impact on this. Moreover, we used a simplified version of HAREM, which originally is a much richer collection.

Anyway, with our pipeline and the integrated CRF module, we hope that more people will have access to an easy-to-use tool for the computational processing of Portuguese, namely in Python, and that interesting applications may come from this. We will, for sure, make further experiments, such as analysing the impact of different feature sets, among those used.

We should also refer that, once we found that no training data was released for the NER task, nor detailed guidelines, we had several doubts regarding our participation. Although the NE categories targeted in this task are common and broadly used, not everything is always consensual and several specificities need to be clarified. Despite this, the organisers minimised this issue by answering questions through the tasks' Google Group.

Despite the interesting results achieved, our approach to the Relation Extraction task was fairly simple. Relations were identified based on a set of rules applied to the sequence of PoS tags between NEs, bootstrapped from a few examples provided by the organisers.

Although more training examples could be used for generating additional rules, increasing coverage and, thus, performance, we are not sure whether the relations covered in this task are the most useful for us. Information extraction has the goal of acquiring meaningful data, while the majority of the relation types considered (e.g., 'de', '(' or 'em') are too vague, difficult to interpret and, thus, hardly useful.

Having this in mind, it is in our plans to develop new versions of FactPyPort, following alternative approaches, covering Open Information Extraction [7], or training a CRF for specific types of relation, as others did for Portuguese [4], possibly using available data (e.g., by Batista et al. [2]). Future versions will, of course, be integrated in NLPyPort.

Acknowledgements

This work was supported by FCT's INCoDe 2030 initiative, in the scope of the demonstration project AIA, "Apoio Inteligente a Empreendedores (Chatbots)".

References

1. Collovini de Abreu, S., Vieira, R.: Relp: Portuguese open relation extraction. Knowledge Organization **44**(3), 163–177 (2017)

2. Batista, D.S., Forte, D., Silva, R., Martins, B., Silva, M.J.: Extracção de relações semânticas de textos em português explorando a dbpédia e a wikipédia. Linguamática **5**(1), 41–57 (2013)
3. Bird, S., Loper, E.: NLTK: The Natural Language Toolkit. In: Proceedings of the ACL 2004 Interactive Poster and Demonstration Sessions. pp. 214–217. ACL (2004)
4. Collovini, S., Machado, G., Vieira, R.: A sequence model approach to relation extraction in Portuguese. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). pp. 1908–1912. ELRA, Portorož, Slovenia (May 2016)
5. Collovini, S., Pereira, B., dos Santos, H.D.P., Vieira, R.: Annotating relations between named entities with crowdsourcing. In: Natural Language Processing and Information Systems - 23rd International Conference on Applications of Natural Language to Information Systems, NLDB 2018. pp. 290–297. Paris, France (2018)
6. Collovini, S., Santos, J., Consoli, B., Terra, J., Vieira, R., Quaresma, P., Souza, M., Claro, D.B., Glauber, R., a Xavier, C.C.: Portuguese named entity recognition and relation extraction tasks at IberLEF 2019. In: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019). CEUR Workshop Proceedings, CEUR-WS.org (2019)
7. Etzioni, O., Fader, A., Christensen, J., Soderland, S., Mausam: Open information extraction: The second generation. In: Proceedings of 22nd International Joint Conference on Artificial Intelligence. pp. 3–10. IJCAI 2011, IJCAI/AAAI, Barcelona, Spain (2011)
8. Ferreira, J., Gonçalo Oliveira, H., Rodrigues, R.: Improving NLTK for processing Portuguese. In: Symposium on Languages, Applications and Technologies (SLATE 2019). pp. 18:1–18:9. OASIcs, Schloss Dagstuhl (June 2019)
9. Freitas, C., Carvalho, P., Gonçalo Oliveira, H., Mota, C., Santos, D.: Second HAREM: advancing the state of the art of named entity recognition in Portuguese. In: Proceedings of 7th International Conference on Language Resources and Evaluation. LREC 2010, ELRA, La Valleta, Malta (May 2010)
10. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of 14th Conference on Computational Linguistics. pp. 539–545. COLING 92, ACL Press, Morristown, NJ, USA (1992)
11. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proc. 18th International Conference on Machine Learning. pp. 282–289. ICML '01, Morgan Kaufmann (2001)
12. Lopes, F., Teixeira, C., Gonçalo Oliveira, H.: Named entity recognition in portuguese neurology text using crf. In: Proceedings of 19th EPIA Conference on Artificial Intelligence. p. In press (September 2019)
13. Okazaki, N.: CRFsuite: a Fast Implementation of Conditional Random Fields (CRFs) (2007), `http://www.chokkan.org/software/crfsuite/`
14. Pantel, P., Pennacchiotti, M.: Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In: Proceedings of 21st International Conference on Computational Linguistics and 44th annual meeting of the Association for Computational Linguistics. pp. 113–120. ACL Press, Sydney, Australia (2006)
15. Pires, A.R.O.: Named Entity Extraction from Portuguese Web Text. Master's thesis, Faculdade de Engenharia da Universidade do Porto (2017)
16. Rodrigues, R., Gonçalo Oliveira, H., Gomes, P.: NLPPort: A Pipeline for Portuguese NLP. In: Proceedings of 7th Symposium on Languages, Applications

and Technologies (SLATE '18). pp. 18:1–18:9. OpenAccess Series in Informatics, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany (June 2018)

17. Santos, D., Seco, N., Cardoso, N., Vilela, R.: HAREM: An advanced NER evaluation contest for Portuguese. In: Proceedings of 5th International Conference on Language Resources and Evaluation (LREC'06). ELRA, Genoa, Italy (May 2006)