RACAI's System at PharmaCoNER 2019

Radu Ion
Institute for AI
Romanian Academy
13 "Calea 13 Septembrie"
Bucharest 050711, Romania
radu@racai.ro

Vasile Florian Păiș
Institute for AI
Romanian Academy
13 "Calea 13 Septembrie"
Bucharest 050711, Romania
vasile@racai.ro

Maria Mitrofan
Institute for AI
Romanian Academy
13 "Calea 13 Septembrie"
Bucharest 050711, Romania
maria@racai.ro

Abstract

This paper describes the Named Entity Recognition system of the Institute for Artificial Intelligence "Mihai Drăgănescu" of the Romanian Academy (RACAI for short). Our best F1 score of **0.84984** was achieved using an ensemble of two systems: a gazetteer-based baseline and a RNN-based NER system, developed specially for PharmaCoNER 2019. We will describe the individual systems and the ensemble algorithm, compare the final system to the current state of the art, as well as discuss our results with respect to the quality of the training data and its annotation strategy. The resulting NER system is language independent, provided that language-dependent resources and preprocessing tools exist, such as tokenizers and POS taggers.

1 Introduction

Named entity recognition (NER) efforts present two challenges: entity detection, identifying the portion of text associated with an entity, and disambiguation, assigning the identified text to a specific entity class. At the Institute for Artificial Intelligence "Mihai Drăgănescu" of the Romanian Academy, one of the research goals focuses on constructing an improved named entity recognition system for Romanian language, including biomedical entities. In this context, the current PharmaCoNER 2019 competition (Gonzalez-Agirre et al., 2019) offered the opportunity to reconsider the existing Romanian NER system which provided the grounds for developing new approaches that are language-independent and more accurate. With respect to our current Romanian biomedical NER system, Mitrofan (2017) presents a neural network based NER system that is able to detect the beginnings and insides of entities with four labels: anatomical parts, disorders, medical procedures and chemical compounds. Although we did not have the time to train this NER system on PharmaCoNER 2019 data, our F1 score on Spanish, when compared to the reported F1 score for the Romanian chemical compounds (the label that best overlaps with the labels of PharmaCoNER 2019), is a strong indicator that we can greatly improve the Romanian biomedical NER system (by how much is the subject for a future paper).

We begin by looking at state of the art approaches for NER systems, presented in Section 2 "Related work", then we continue with the resources used for this specific task, in Section 3 "Resources", followed by a presentation of our implemented algorithms and methods, in Section 4 "RACAI Systems". Finally, system evaluation results are presented in Section 5 "System evaluation", followed by conclusions.

2 Related work

To tackle the challenges posed by BioNER, different NER approaches were proposed. Even though high performances have been obtained by applying classical NER approaches such as dictionary-based methods (Sekine and Nobata, 2004), rule-based methods (Rau, 1991), Hidden Markov Models (Zhou and Su, 2002), Conditional Random Fields (Dingare et al., 2005), the current dominant techniques are based on neural methods, which will also be our focus in this paper, mainly because we think that this is the current state of the art approach to NER.

Deep learning methods have shown impressive results when applied to NLP and, since (Hochreiter and Schmidhuber, 1997) proposed Long-Short Term Memory neural networks and Bidirectional Long-Short Term Memory (BiLSTM) networks (Graves, 2012), a wide variety of NER systems have been created based on these methods.

Proceedings of the 5th Workshop on BioNLP Open Shared Tasks, pages 90–99
Hong Kong, China, November 4, 2019. ©2019 Association for Computational Linguistics

Santos and Guimaraes (2015) presented a language-independent approach for NER based on a deep neural architecture that uses word and character-level embeddings to perform sequential classification. In order to demonstrate the language-independence of the system, two annotated corpora in two different languages were used: a Portuguese corpus - HAREM I (Milidiú et al., 2008) and a Spanish corpus - SPA CoNLL-2002 (Sang and F., 2002). The system obtained an F1 score of 79% when trained on HAREM I corpus and an F1 score of 82.2% for the SPA CoNLL-2002 corpus.

Chiu and Nichols (2016) presented a NER system based on stacked BiLSTM architecture trained to detect four types of entities such as: "PERSON", "ORGANIZATION", "LOCATION" and "MISC", each of the entity being annotated in BIOES format (Beginning, Inside, Outside, Ending and Single). Using two lexicons extracted from publicly-available resources the system obtained an F1-score of 91.62% on CoNLL-2003 (Sang and De Meulder, 2003) corpus and 86.28% on OntoNotes (Pradhan et al., 2013) corpus.

Shao et al. (2016) evaluated the performances of three types of neural networks based systems for multilingual NER. They compared a windows-based feed-forward network, a standard BiLSTM and a window-based BiLSTM. Word embeddings combined with word-level features were used and the annotation format was also BIOES. Based on the experiments the authors concluded that: the feed-forward neural network was outperformed in accuracy by the standard BiLSTM and when less information is available, the window-based BiLSTM is more robust than the standard BiLSTM.

Soares et al. (2019) used NeuroNER (Dernoncourt et al., 2017) framework in order to perform NER for medical domain. The Spanish Clinical Cases Corpus (SPACCC) was used to train the system, which is based on a LSTM neural network. The biomedical corpus was previously annotated with four entity types, a subset of the types PharmaCoNER 2019 uses. Using medical word-embeddings, the system achieved an F1 score of 88.18%, outperforming the baseline system which scored 87.76%.

3 Resources

In order to develop, train and test a NER system several resources are needed. In this section we review the main types of linguistic resources used in our work:

3.1 Corpora

When applied to general domain, most of the state of the art systems make use of the CoNLL-2002 corpus (Sang and F., 2002), which contains six files that cover two languages: Dutch and Spanish. The set of entity labels used for this corpus contains four types of entities: PER (persons), ORG (organizations), LOC (locations) and MISC (miscellaneous).

In order to perform named entity recognition on biomedical textual data several annotated corpora were developed. For English there are several annotated corpora used for biomedical NER such as: **NCBI** (Doğan et al., 2014) a gold-standard corpus for disease mentions and concepts that contains 793 abstracts extracted from PubMed; **CHEMDNER** (Krallinger et al., 2015) a corpus of 10,000 abstracts collected from PubMed annotated with two types of NEs: chemicals and drugs.

Lately a slightly increasing number of resources specific to this field have been created for languages other than English. For example for French there is the **Quaero** corpus (Névéol et al., 2014) which contains 103,056 words annotated with ten types of NEs defined using UMLS: anatomy, chemical and drugs, devices, disorders, geographic areas, living beings, objects, phenomena, physiology, procedures. For Romanian there is the **MoNERo** (Maria Mitrofan, 2019) corpus which is a biomedical gold standard corpus and contains 154,825 words annotated with four types of entities: anatomy, chemicals and drugs, disorders and procedures. For Spanish **IxaMedGS** (Oronoz et al., 2015) is a corpus that contains 142,154 discharge records out of which 75 were annotated with two types of NEs: diseases and drugs; **DrugSemantics** corpus (Moreno et al., 2017) has 226,729 tokens annotated with ten types of NEs: chemical composition, disease, drug, excipient, food, medicament, pharmaceutical form, route, therapeutic action and unit of measurement.

3.2 Word embeddings

Continuous word representations, trained on large corpora have been proven to be useful for many NLP tasks, including NER. It is known that neural word representations have the ability to capture useful semantic properties and linguistic relationships between words (Bakarov, 2018). Therefore pre-trained word embeddings are available for different languages, including Romanian and Spanish. For example in Romanian we have a set of word embeddings (Păiș and Tufiș, 2018) computed on the Reference Corpus for Contemporary Romanian Language (CoRoLa) (Barbu Mititelu et al., 2018) corpus.

Grave et al. (2018) released a set of pre-trained embeddings for 157 languages calculated on texts extracted from Wikipedia. Also for Spanish there is a different set of pre-trained embeddings made available by the Chile NLP group[1] and calculated using the Spanish Billion Word Corpus (SBWCE)[2].

Chiu and Nichols (2016) showed that word embeddings vectors calculated on a specific domain produce better results than those obtained from general-domain texts. Therefore (Soares et al., 2019) calculated a set of medical word embeddings for Spanish. They used text from two sources: full medical articles from SciELo database[3] (100 million tokens) and biomedical texts from Wikipedia (82 million tokens). The experiments performed using this resource generated more accurate results than those calculated based on general-domain texts.

3.3 SNOMED CT

SNOMED CT (Systematized Nomenclature of Medicine - Clinical Terms)[4] is a multilingual healthcare terminology built around a concept-based ontology. It contains more than 1 million distinct medical terms, 326,734 concepts and 19 hierarchies. Concepts are classified under hierarchies, of which most of them corresponding to the types of entities instances of which are encountered by clinicians during their work (body parts, diseases, substances, procedures, etc.). A concept in SNOMED CT has a unique name, unique numeric code, and more descriptions (one main definition, several secondary and more synonyms). This resource is available in both English and Spanish. To use it for scientific purposes, a license is required after completing a form. We used this resource to extract all the available proteins and genes. Using the SNOMED browser for Spanish[5] we extracted 9,556 proteins names.

4 RACAI Systems

4.1 RACAI Baseline

Our baseline system is an enhanced gazetteer-based annotation tool. It takes as input multiple files, each containing an entity list of the same type. For example, in the `PROTEINAS.txt` file there will be a list of proteins. On each line, there will be a string containing a word or an expression denoting a protein.

Various gazetteer annotation systems already exist. We recall here Stanford TokensRegex (Chang and Manning, 2014) and Stanford RegexNER part of Stanford Core NLP (Manning et al., 2014). However, these and similar other systems, impose the format to correspond to some specific regular expression syntax (or at least to a certain fixed form textual representation). In our case, the gazetteer resources are partially generated directly from the training annotations provided for the task. Therefore, the format used is not directly checked and validated by a human operator.

Therefore, our system does not look for expressions exactly as they are provided. Instead it implements additional rules to improve matching such as:

- ignore special characters (example: ‘-’ , ‘¡’, ‘(’ etc.) in both provided expressions and the searched text;
- recognize words followed by numbers regardless of the way they are written (for example: “CAP-57”, “CAP 57”, “CAP57”).

Finally, in the case of overlapping entities being found, the longer one is kept. The software program allows for such overlapping entities to be saved for manual examination, but this particular feature did not seem useful for this task. The resulting annotation file is in the “.ann” format.

4.2 RPCN

RPCN stands for the “RACAI PharmaCoNER neural network” and is, as its name suggests, a neural network that we specifically designed for this competition and that, ultimately, will also be run for Romanian for which we have BioNER training data (Maria Mitrofan, 2019).

[1] https://github.com/dccuchile/spanish-word-embeddings

[2] http://crscardellino.github.io/SBWCE/

[3] https://www.scielo.org/en/

[4] https://www.snomed.org/snomed-ct

[5] https://browser.ihtsdotools.org

4.2.1 Comparison with the state of the art and design choices

As already discussed in Section 2, NER systems based on BiLSTMs and using convolutional neural networks (CNNs) to encode character-based features of the input (Chiu and Nichols, 2016) represent the current state of the art for NER task. Other approaches used stacked BiLSTM layers in an attempt to increase the generalization power of the network or decoders which chose the most probable label output given the LSTM encoding of the featurized input (Dernoncourt et al., 2017).

Our research goal was to test an approach based on BiLSTMs, given the abundance of papers using this type of artificial cell and reporting very good results. At this point, we have to mention that *all design choices of RPCN presented below were driven by intense experimentation* with the provided training data, aiming at *short training and evaluation* loops. Because the training data is rather small in size (a bit more than 3800 training examples), we quickly realized that running with more complex architectures (which have more parameters) leads to overfitting. Thus, all architectures with two BiLSTM layers and/or CNNs encoding character features were dropped early on from our experiments.

The RPCN network differs by mainstream BiLSTM NER networks by attempting to use an attention mechanism, like the one in (Anh Nguyen et al., 2019) (of which we did not know at the time of our experiments), whose main function is to model how much words surrounding labeled entities contribute to the label prediction. Also, RPCN tries to combine (by a simple addition) independently trained word embeddings from the medical domain with the embeddings extracted directly from the training corpus. We found that this approach gives a significant boost of performance (more than 10% in the F1 score) when compared to the usage of either word embedding sources in isolation or with general-purpose embeddings extracted from Wikipedia. We are thus able to confirm and supplement the findings of Soares et al. (2019).

In relation to the featurized input that we designed for RPCN, we were guided by the following assumptions and intuitions:

- all NEs are mostly noun phrases and in Spanish, as in Romanian, noun phrases have a well-defined syntactic structure which prompted the usage of POS tags as features;
- all NEs are medical substances obeying some naming patterns, so a feature regarding words affixes was needed;
- some proteins have specific character patterns, so a "word shape" feature was also thought to be useful (see the next subsection for the "shape features" details);
- with an eye to the rank of our system in the PharmaCoNER 2019 competition, we also thought that including the gazetteer feature (if available) directly into RPCN would increase the performance of the system.

4.2.2 Architecture

RPCN is a RNN which uses LSTM cells to encode the feature descriptions of the words coming in, remembering the information from both left and right contexts of the target word, which makes it BiLSTM RNN. The network was trained to label each word in the sequence with one of the PharmaCoNER target labels or with the "nothing interesting here" label which we called `NONE`.

The RPCN architecture is presented in Figure 1. We have tried the vanilla variant and the variant enhanced with an attention mechanism, as described by Bahdanau et al. and retained the latter for further development, as the better approach. RPCN is written in Java 1.8, using the DeepLearning4J deep neural network Java library, version 1.0.0-beta3.

Figure 1 shows the input vectors and the BiLSTM cell for a single input word, for example `cadenas`, but we consider sequences of words, each with its own BiLSTM cell (but shared parameters among words). The input vectors that go into the BiLSTM cell are as follows:

- the `WE Layer` is the word embedding layer for the input word; its output size was chosen by our hyperparameter grid search procedure to be 64 (see the Training subsection 4.2.3). The word is one-hot encoded and fed to this layer which compresses it to a 64 dimensional vector;
- the `External WEs` resource refers to our pretrained Spanish medical word embeddings (Soares et al., 2019). Because the size of these embeddings is larger than 64, one such embedding is fed to a fully connected

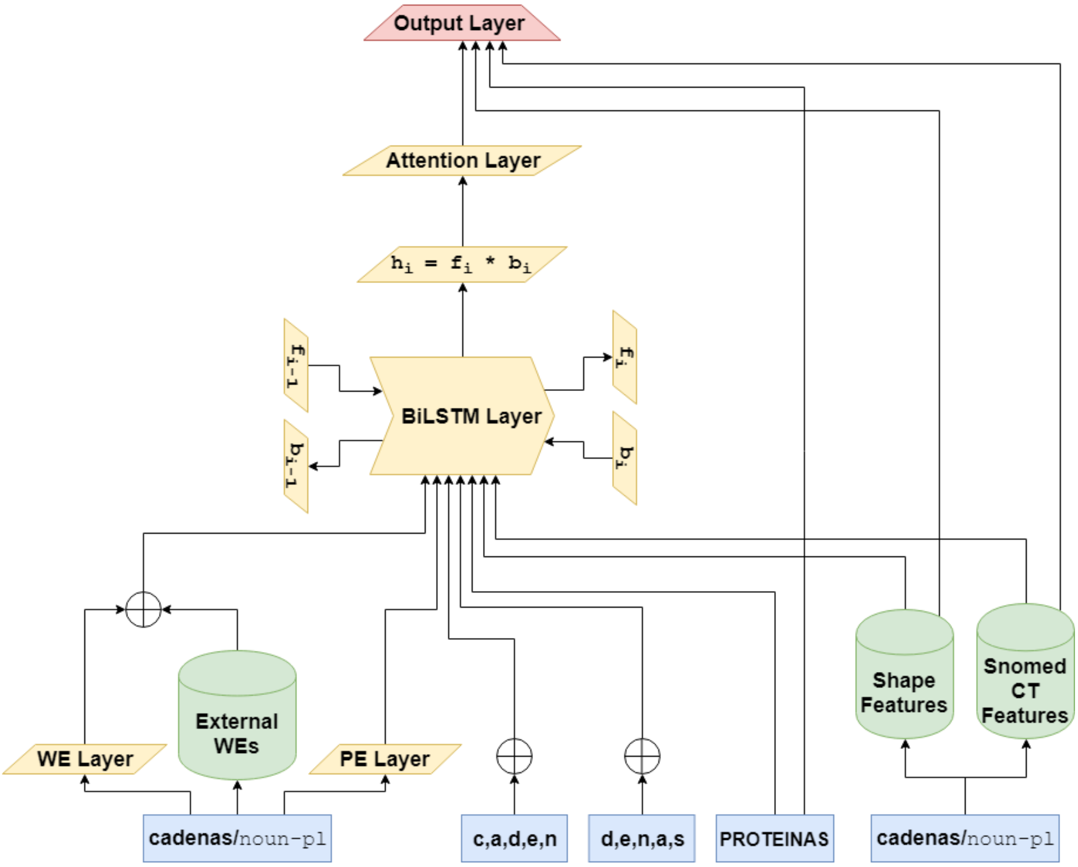

Figure 1: The RPCN neural network

layer with an output size of 64 so that we can add (element-wise) the output of the `WE Layer` with the output of this fully connected layer to obtain a "unified" word embedding representation for the input word;

- the `PE Layer` is the POS tag embedding layer for the POS tag of the input word (a plural noun for our example word); its output size was chosen to be 16 by the hyperparameter grid searching procedure. Each POS tag is encoded as a one-hot vector and fed to this layer which compresses it to a 16 dimensional vector;

- `c,a,d,e,n` and `d,e,n,a,s` are the "relative-index-hot" representations of the 5 character prefix and suffix of the input word; the vectors for each character are added to form a single output vector. The "relative-index-hot" stands for the use of the $1/(i+1)$ quantity instead of a 1 on the corresponding vector position, where i is the index of the character in the input word (0-based numbering), and this trick allows us to encode in a single vector both the prefix and suffix vectors which are sensitive to the character ordering in the word;

- `PROTEINAS` is the one-hot representation of the gazetteer label that is (optionally) available for the input word (if it is not available, we use the the one-hot representation of the "default" label `NONE`);

- the `Shape Features` resource refers to our word shape extraction algorithm that does the following:

  - using regular expressions, sets one bit in the feature binary vector if the input word looks like a substance, e.g. "CD34", "CAM5.2", "Tc99m-MDP", etc.
  - sets one bit in the the feature binary vector if the word is a "dash prefix word", e.g. "alfa", "Beta", "β", etc. and it is "glued" (no spaces) to the next word; the list of dashed prefix words has been automatically generated from the train set.

- the `SNOMED CT Features` resource refers to our Spanish SNOMED CT "word as a feature" algorithm. Based on our tokenized SNOMED CT gazetteer list, in which we labeled each "(sustancia)" concept description with either the `PROTEINAS` or the `NORMALIZABLES` labels, we counted each word and label pair and then computed the probability distribution $P(\texttt{PROTEINAS}|word)$ and $P(\texttt{NORMALIZABLES}|word)$. Each word is represented by a 2 float vector on how probable it is to point to either of these two labels. If the input word is not found in this resource, $P(\texttt{PROTEINAS}|word) = P(\texttt{NORMALIZABLES}|word) = 0.5$. To the 2 float vector we also append the relative frequency of the word in the tokenized SNOMED CT gazetteer list; naturally, we skip functional words from this computation. Some example vectors, computed as described above, are presented in Table 1.

The BiLSTM cell will combine the forward f_i and backward b_i states by multiplying the state vectors, element-wise: $h_i = f_i \cdot b_i$. This method proved to increase the precision of the system as the signal will be strong only if left and right evidence is strong (i.e. close to 1.0). We also found out that if we average the forward and backward states as in $h_i = (f_i + b_i)/2$, we can increase the recall at the cost of a lower precision. The same effect (recall increase) is obtained when the forward and backward states are concatenated.

Besides the weighted sum of the combined BiLSTM outputs h_i given by the attention layer, the output layer (a `softmax` layer with the output size equal to the number of target labels) also receives the raw inputs from the gazetteer feature, the shape features and the SNOMED CT features, in an effort to boost the precision of the system.

4.2.3 Training

The input text is tokenized first, using an in-house built tokenizer for Spanish, specifically designed for this task. The tokenizer will split words at the dash ('-') boundary because we observed that some entities contained the dash while others did not. The tokenizer will recognize (and thus generalize) the following types of tokens: numbers (integers, reals, Roman numerals), amounts (e.g. "305mg"), units of measure (e.g. "mg/g"), temperatures (e.g. "30°C") and area/volume expressions (e.g. "3x2cm2"). After tokenization, the text is POS tagged using the Stanford Core NLP suite with the Spanish POS tagging model and the sentence boundaries are detected using a simple regular expression: end of sentence punctuation followed by whitespace and then by an uppercase letter. No named entity is allowed to cross a sentence boundary.

We used a grid searching procedure, together with the supplied train and development data, to optimize the hyperparameters of RPCN. The hyperparameters are as follows:

- the *number of time steps* in the sequence: how many words are in a window of consecutive words that the RPCN can consider as a training example. Tried values were in the set $\{7, 11, 15, 19, 21, 25\}$ and the best value was set to 21;

- the *size of the LSTM state* vector; tried values were in the set $\{64, 128, 256\}$ and the best value was set to 128;

- the *size of the trained word embedding* vector, i.e. the size of the `WE Layer`. Tried values were in the set $\{32, 64, 128\}$ and the best value was set to 64;

- the *size of the POS tag embedding* vector, i.e. the size of the `PE Layer`. Tried values were in the set $\{8, 16\}$ and the best value was set to 16.

The train and development sets that were made available by the task organizers were distributed as follows: 3822 training annotations (T entries in the ".ann" files) and 1926 development annotations. We have randomly reshuffled the whole data set (training plus development) into 90% training set and 10% development set.

As far as the configuration of the computation graph goes, we used the Xavier weight initialization method together with the Stochastic Gradient Descent optimization algorithm and the Adam updater with the default parameters. The reader can refer to the documentation of the DeepLearning4J library for a description of these methods.

4.2.4 Running

The incoming text is tokenized, POS tagged and sentence split. Then, RPCN is run on consecutive sequences of adjacent words of length 21, each

Word	$P(\texttt{PROTEINAS}\|word)$	$P(\texttt{NORMALIZABLES}\|word)$	$P(word)$
lormetazepam	0.0	1.0	8.379841E-6
antinuclear	0.5	0.5	1.005581E-4
oxigenasa	0.625	0.375	6.703873E-5
carveol	1.0	0.0	8.379841E-6

Table 1: SNOMED CT word features for labels PROTEINAS and NORMALIZABLES

word receiving the best label by the softmax output, accumulating labels as the window passes by. The label with the highest accumulated score wins for each word. Spans of consecutive tokens having the same non-NONE labels are the new detected named entities.

The raw label assignments are post-processed to enforce the following:

- a recognized named entity will not start or end with a functional word;
- if there is a gazetteer annotation for a RPCN detected span then the labels must agree and the gazetteer span boundaries will be preferred. If the labels do not agree, both spans are deleted.

Finally, we also apply some regular expression based rules to catch some expressions which RPCN was not able to learn, e.g. CD[0-9]+ (a protein) or the pattern W1 "de" W2 in which "de" W2 receive the same label as W1.

4.3 Ensemble methods

Given the different annotator systems described above, an ensemble system was needed. Its aim was to take the resulting annotations from two or more runs, with the same or different system, and combine them using different rules in order to improve the overall results. The idea behind it is that each system could be better at detecting certain types of entities and the combined annotation would be better overall.

Our combining system takes as input two ".ann" files and produces another ".ann" file by applying rules. The rules are especially useful in the case of overlapping entities. If there are no overlapping entities, then the input annotations are simply merged. Currently there are 5 rules available:

- "PRIO1": gives priority to the first input file, retaining the corresponding entity annotation;
- "PRIO2": gives priority to the second input file;
- "SMALLER": keeps the smaller annotation, discarding the longer one in case of entity overlap;
- "LARGER": keeps the longer annotation;

5 System Evaluation

5.1 Working methodology

We mentioned that the initial distribution of annotations in the training and development sets was not satisfactory and thus, we have proceeded to the random reshuffling of the whole data set followed by a 90%/10% split. We have selected our best ensemble method on such a random reshuffling and training/development split.

The RACAI baseline system worked with the annotations from the training set plus the gazetteer list based on the Spanish SNOMED CT "(sustancia)" concept descriptions which we automatically extracted and labeled as either PROTEINAS or NORMALIZABLES and then manually validated.

RPCN was trained on the training set and evaluated, along with the RACAI baseline system, on the development set. For the *official evaluation run*, we used all annotations from the provided data set and the SNOMED CT entries as the gazetteer list.

5.2 Results

Table 2 presents the runs of the RACAI baseline system, RPCN and of four ensemble methods applied to the baseline (first input) and RPCN (second input).

The highest scores are bold-faced for the Precision (P), Recall (R) and F1 columns. According to our evaluations, the best ensemble method (by the F1 score which was the optimization target) is the "LARGER" (or C4 to match the name of the submitted zip file) ensemble method. Knowing that we are allowed to submit five different runs, based

System	P	R	F1
Baseline	0.8986	0.6915	0.7816
RPCN	**0.9025**	0.7539	0.8215
PRIO1 (C1)	0.8733	0.7764	0.8220
PRIO2 (C2)	0.8871	0.7764	0.8281
SMALLER (C3)	0.8694	0.7730	0.8183
LARGER (C4)	0.8911	**0.7799**	**0.8318**

Table 2: Development results of RACAI's NER systems

System	P	R	F1
Baseline	**0.92530**	0.71281	0.80527
RPCN	0.89327	0.76330	0.82319
PRIO1 (C1)	0.90189	0.80347	**0.84984**
LARGER (C4)	0.90043	0.79533	0.84462
C4M	0.78281	**0.84528**	0.81284

Table 3: Official PharmaCoNER 2019 results of RACAI's NER systems

on these evaluations, we decided to submit the output of the following systems: RPCN (best precision), LARGER (C4, best F1 score) and Baseline (official reference system). Before the submission deadline, we also sent the PRIO1 (C1, the Baseline priority) and an ensemble between the Baseline and one other system that we developed for PharmaCoNER 2019 (C4M). Table 3 presents the official results that were communicated to us by the task organizers.

6 Discussion and conclusions

The official evaluation results confirmed the results we obtained during development: the PRIO1 and LARGER ensembles between the Baseline and the RPCN systems are better than each of them, individually. RPCN definitely learned to recognize new entities, as its recall is larger with more than 5% than the recall of the Baseline system.

We can also see that the precision of RPCN dropped, as compared to the precision of the Baseline system, with more than 3% in the official evaluation. This discrepancy appeared during development as well and the main reason we found for it was that the training data *was not consistently annotated*. That is, the same expression (same words, same casing) was annotated in a document and was not annotated in another document. We do not think that at this specialization level we can justify this at a semantic level (i.e. the expression does not mean the same thing in the two documents). Thus, during development, we automatically re-annotated the whole supplied data, making sure the same expression is annotated everywhere with the same label (if there was an ambiguity, the re-annotation was cancelled for the expression). By doing this, we were able to close the precision gap between the Baseline and the RPCN systems.

While we do not know the rank of our system yet, our best system was scored with an F1 score of **0.84984**, which, we feel, is good performance. We will put this system to the tests of scalability and language-independence by using it unchanged (but with the specialized computational resources) in two Romanian-related tasks: as already stated, in the identification of Romanian biomedical NEs and in the rather different task of legal terminology identification (e.g. EuroVoc[6]) in Romanian legal texts, to be performed in the MARCELL project[7]. For the latter task, we will have the chance to determine if our system is able to reliably detect new terms which are missing from the legal terminology dictionaries.

Acknowledgments

The reported research was supported by the EC grant MARCELL (Multilingual Resources for CEF.AT in the Legal Domain), TENtec no. 27798023.

References

Kim Anh Nguyen, Ngan Dong, and Cam-Tu Nguyen. 2019. Attentive Neural Network forNamed Entity Recognition in Vietnamese. *2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF)*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR 2015*.

Amir Bakarov. 2018. A Survey of Word Embeddings Evaluation Methods.

Verginica Barbu Mititelu, Dan Tufiș, and Elena Irimia. 2018. The reference corpus of contemporary Romanian language (CoRoLa). In *Proceedings of the 11th Language Resources and Evaluation Conference – LREC 2018*, pages 1178–1185, Miyazaki,

[6] https://data.europa.eu/euodp/en/data/dataset/eurovoc

[7] http://marcell-project.eu/

Japan. European Language Resources Association (ELRA).

Angel X. Chang and Christopher D. Manning. 2014. Tokensregex: Defining cascaded regular expressions over tokens.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Franck Dernoncourt, Ji Young Lee, and Peter Szolovits. 2017. Neuroner: an easy-to-use program for named-entity recognition based on neural networks. *arXiv preprint arXiv:1705.05487*.

Shipra Dingare, Malvina Nissim, Jenny Finkel, Christopher Manning, and Claire Grover. 2005. A system for identifying named entities in biomedical text: how results from two evaluations reflect on both the system and the evaluations. *International Journal of Genomics*, 6(1-2):77–85.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

Aitor Gonzalez-Agirre, Montserrat Marimon, Ander Intxaurrondo, Obdulia Rabal, Marta Villegas, and Martin Krallinger. 2019. Pharmaconer: Pharmacological substances, compounds and proteins named entity recognition track. In *Proceedings of the BioNLP Open Shared Tasks (BioNLP-OST)*, pages 1–X, Hong Kong, China. Association for Computational Linguistics.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Alex Graves. 2012. Supervised sequence labelling. In *Supervised sequence labelling with recurrent neural networks*, pages 5–13. Springer.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M Lowe, et al. 2015. The chemdner corpus of chemicals and drugs and its annotation principles. *Journal of cheminformatics*, 7(1):S2.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Prismatic Inc, Steven J. Bethard, and David Mcclosky. 2014. The Stanford CoreNLP Natural Language Processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Grigorina Mitrofan Maria Mitrofan, Verginica Barbu Mititelu. 2019. Monero: a biomedical gold standard corpus for the romanian language. In *Proceedings of 18th ACL Workshop on Biomedical Natural Language Processing*, volume (In press).

Ruy Luiz Milidiú, Cícero Nogueira dos Santos, and Julio Cesar Duarte. 2008. Portuguese corpus-based learning using etl. *Journal of the Brazilian Computer Society*, 14(4):17–27.

Maria Mitrofan. 2017. Bootstrapping a Romanian Corpus for Medical Named Entity Recognition. In *Proceedings of Recent Advances in Natural Language Processing*, pages 501–509, Varna, Bulgaria.

Isabel Moreno, Ester Boldrini, Paloma Moreda, and M Teresa Romá-Ferri. 2017. Drugsemantics: a corpus for named entity recognition in spanish summaries of product characteristics. *Journal of biomedical informatics*, 72:8–22.

Aurélie Névéol, Cyril Grouin, Jeremy Leixa, Sophie Rosset, and Pierre Zweigenbaum. 2014. The quaero french medical corpus: A ressource for medical entity recognition and normalization. In *In Proc BioTextM, Reykjavik*. Citeseer.

Maite Oronoz, Koldo Gojenola, Alicia Pérez, Arantza Díaz de Ilarraza, and Arantza Casillas. 2015. On the creation of a clinical gold standard corpus in spanish: Mining adverse drug reactions. *Journal of biomedical informatics*, 56:318–332.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152.

Vasile Păiș and Dan Tufiș. 2018. Computing distributed representations of words using the COROLA corpus. *Proceedings of the Romanian Academy, Series A*, 19(2):403–409.

Lisa F Rau. 1991. Extracting company names from text. In *[1991] Proceedings. The Seventh IEEE Conference on Artificial Intelligence Application*, volume 1, pages 29–32. IEEE.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. corr. *arXiv preprint cs.CL/0306050*.

Tjong Kim Sang and Erik F. 2002. Introduction to the CoNLL-2002 shared task. *Proceedings of the 6th Conference on Natural Language Learning - COLING-02*.

Cicero Nogueira dos Santos and Victor Guimaraes. 2015. Boosting named entity recognition with neural character embeddings. *arXiv preprint arXiv:1505.05008*.

Satoshi Sekine and Chikashi Nobata. 2004. Definition, dictionaries and tagger for extended named entity hierarchy. In *LREC*, pages 1977–1980. Lisbon, Portugal.

Yan Shao, Christian Hardmeier, and Joakim Nivre. 2016. Multilingual named entity recognition using hybrid neural networks. In *The Sixth Swedish Language Technology Conference (SLTC)*.

Felipe Soares, Marta Villegas, Aitor Gonzalez-Agirre, Martin Krallinger, and Jordi Armengol-Estapé. 2019. Medical word embeddings for spanish: Development and evaluation. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 124–133.

GuoDong Zhou and Jian Su. 2002. Named entity recognition using an hmm-based chunk tagger. In *proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 473–480. Association for Computational Linguistics.