Multi-Input RIM for Named-Entity Recognition in Spanish Clinical Reports

Parsa Bagherzadeh[1], Harsh Verma[1] and Sabine Bergler[1]

[1]CLaC Labs, Concordia University, Montréal, Canada

Abstract

This paper summarizes the CLaC submission for the LivingNER competition which concerns recognition and normalization of named entities in Spanish clinical reports. We integrate diverse external knowledge sources such as BETO (Spanish BERT), UMLS, and POS information using the mi-RIM (Multi-Input Recurrent Independent Modules) architecture. At LivingNER 2022, we obtain a micro-F1 of 93.2 (significantly outperforming the competition mean at 82.3) for Subtask 1 (Named Entity Recognition) and for Subtask 2 (normalization) we obtain a micro-F1 of 91.9 for the normalization task.

Keywords

Multi-input RIM, graph embedding, UMLS, decoupled modules

1. Introduction

Curated health information plays an important role in the prevention and prediction of diseases, improvement of therapies, identification of drug side effects, and consequently, reduction of healthcare costs. The information, however, is buried in a huge amount of raw textual data, and efficient and automatic systems are needed to facilitate access to targeted information for medical practitioners.

Clinical reports, for instance, contain crucial information such as personal history, family background, or living environment, which are all important for medical practitioners to make decisions for treatment of patients with similar conditions. This information need motivated the LivingNER challenge [1], which comprises a corpus of 1000 Spanish clinical reports. Subtask 1 is a Named Entity Recognition (NER) task which asks to recognize and classify living things into the two categories HUMAN and SPIECIES. Subtask 2 requires normalizing each entity detected in Subtask 1 to the corresponding NCBI tax code.

Example 1 shows a clinical report from the LivingNER data, where entities that the goldstandard annotates as HUMAN are underlined, and those annotated as SPECIES are doubly underlined.

(1) *ANTECEDENTES FAMILIARES Padre ingresado desde hace 15 días por sospecha de tuberculosis pulmonar (clínica respiratoria de tos y febrícula de 6 meses de evolución). Afectación en radiografía de tórax y TAC pulmonar y Mantoux positivo. Ha iniciado tratamiento antituberculoso desde hace 10 días. Mantoux a contactos familiares pendientes de leer. Resto sin interés.*

IberLEF 2022, September 2022, A Coruña, Spain.

Clinical reports include specialized terms, which makes the NER task challenging for models pre-trained on general language, particularly if the terms are not sufficiently foreshadowed in the training data. To counteract this issue, ontological resources such as UMLS [2] can be leveraged to improve the coverage of a NER system. UMLS includes several vocabularies, including Spanish lexica.

Example 2 shows a data sample with three separate entities of interest. Example 3, however, includes a case of overlapping annotations. This occurs mostly for SPECIES. Such overlapping entities are often listed mentions of different variants of the same virus/bacterium.

(2) *Serología en LCR a Mycoplasma pneumoniae, Virus de Epstein-Barr, Citomagalovirus, …*

(3) *Como hallazgo casual se observa una seroconversión del virus de la hepatitis A y E, sin repercusión clínica …*

We address the LivingNER task using the mi-RIM (multi-input Recurrent Independent Mechanisms) architecture [3]. mi-RIM comprises a set of interacting recurrent modules that are independent in their dynamics. We use different knowledge sources as inputs for different modules. Moreover, we define two objectives for recognizing overlapping and non-overlapping spans. The official results show that the mi-RIM system performs well on the test set, significantly outperforming the competition mean.

2. System

2.1. Pre-processing

We pre-process the data using a GATE pipeline [4]. For tokenization, sentence splitting, and POS tagging we use the Spanish resources in the CoreNLP toolkit (Spanish version).

For our model to predict spans correctly, we need to split hyphenated terms like *anti-VHC* into the tokens [*anti*, -, *VHC*]. This is required because a term like *anti-VHC* contains the SPECIES entity *VHC* and the prefix *anti-* should not be annotated. If the token *anti-VHC* is not split, our model will incorrectly learn to annotate the whole term *anti-VHC* as SPECIES. Figure 1 shows an example of how the CoreNLP tokenizer fails to split hyphenated terms. Hence, we use a simple script to split all hyphenated terms produced by the CoreNLP tokenizer.

CoreNLP tokens:	[…, anti-VHC, - , anti-HBc , + , anti-HBs , +, HBsAg , -, …]
Split at hyphens:	[…, anti , - , VHC, - , anti, -, HBc , + , anti , - , HBs , +, HBsAg , - ,…]

Figure 1: Post-processing after CoreNLP tokenizer

2.2. Knowledge sources

Our system incorporates four extant knowledge sources: BETO as a pre-trained language model, POS, UMLS for authoritative domain terminology, and an ad hoc gazetteer to denote group terms that denotes groups of humans.

BETO To embed tokens we use the Spanish version of BERT (Pre-training of Deep Bidirectional Transformers), called BETO [5], accessed through HuggingFace.

UMLS All terms matched by Spanish entries of UMLS (Unified Medical Language System) [2] are embedded using the adversarially trained graph embeddings provided by [6]. If a term like *Hepatitis A* has more than one match like *Hepatitis* and *Hepatitis A*, we choose the longer match for its embedding. Note that we do not fine-tune the embeddings during the training process.

POS tags POS (Part-of-Speech) information are widely used for NER tasks with proven benefit [7]. Following [8], we pre-train a set of embeddings for POS tags using CBoW (Continuous Bag-of-Words) model [9]. We pre-train the embeddings ($d_{POS} = 20$, window=5), using the Gensim package [10].

Group Gaz mentions of class HUMAN include group terms, such as *nurses*. We compile an ad-hoc gazetteer from UMLS including all terms with a UMLS semantic type *Group*.

2.3. Subtask 1 (NER)

We use the mi-RIM architecture for integration of knowledge sources [3]. The mi-RIM architecture comprises M recurrent modules $f_1, f_2, \ldots, f_M$ that have independent dynamics (no parameter sharing) but they interact via an attention bottleneck. To encourage developing expertise, a limit k can be set that restricts the number of active modules at any time point to k modules. Note that modules may have different inputs. This allows different knowledge sources to be used as the input to different modules.

Because there are a number of annotations for SPECIES, where the annotated strings overlap (see Example 3: *virus de la hepatitis A* and *virus de la hepatitis A y E*), we cast the NER task as two sequence labeling problems. The first task focuses on predicting extended spans, whereas the second task deals with prediction of short spans. We use 6 recurrent modules: f_1 and f_2 for BETO, f_3 and f_4 for UMLS, f_5 for POS, and f_6 for Group Gaz. Among these modules, the hidden states of f_1 and f_3 are used for predicting extended spans, and the hidden states of f_2 and f_4 are used for predicting the short spans. Note that since the modules interact with each other, f_1–f_4 are aware of f_5 and f_6 which are responsible for accommodation of POS and Group Gaz. We jointly train on the two tasks.

2.4. Subtask 2 (Normalization)

Subtask 2 requires normalizing the entities detected in Subtask 1 to the corresponding NCBI tax codes. We use a simple matching approach.

For all entities identified as HUMAN we assign 9606 as default NCBI code which corresponds to *Homo sapiens*. To normalize SPECIES mentions, however, we compile three gazetteer lists G_{Train}, G_{UMLS}, and G_{NCBI}. G_{Train} is populated solely from the training data, which includes training entities matched with their NCBI code. G_{UMLS} contains all Spanish UMLS concepts for which a NCBI code is reported. Finally, G_{NCBI} includes all terms from the NCBI taxonomy. Note that almost all terms in G_{NCBI} are in English, and the gazetteer is intended to capture any possible mentions in English, specifically acronyms. We use these gazetteer lists as follows:

Exact: We match against the three gazetteer lists while ignoring case. First, an entity is matched against G_{Train}. When the gazetteer list returns no hit, we match the entity against G_{UMLS} and G_{NCBI}. If an entity is not matched with any of the three gazetteers, we report the default string "OTHER_CODE".

Levenshtein: The entities that did not match against any gazetteer in the Exact approach above are again matched against G_{Train} by finding the gazetteer entry with the least normalized Levenshtein distance. Normalized Levenshtein distance is a number between 0 and 100. A substitution cost of 2 is used for the calculation. If the normalized distance between two strings is below 15, we report the corresponding NCBI code. Otherwise, we report the default code "OTHER_CODE". The cutoff value was empirically determined on the validation set.

2.5. Implementation details

The mi-RIM model is implemented using the PyTorch library [11]. We train the model using the Adam optimizer [12] with a learning rate of $lr = 5 \times 10^{-6}$. We use early stopping and observe that the best model is usually obtained by the 7th epoch. k is empirically set to 4. $k = 4$ may be optimal because it encourages competition between the six modules while allowing all four knowledge sources to participate, if appropriate.

3. Results

3.1. Development phase

Table 1 reports performance on the development set provided by the organizers. We use the official evaluation script to measure the performances. The baseline shows a rather high performance for Subtask 1, suggesting that the development set is well foreshadowed by the training data. Nevertheless, leveraging UMLS provides a significant boost to the baseline performance. Interestingly, UMLS improves both precision and recall, suggesting that the modules are interoperating well and do not confound each other.

Integration of POS information also improves the performances, albeit marginally. Note that POS seems to improve precision more than recall. This has been also observed in studies such as [8]. Adding the Group Gaz also improves the performance marginally.

The performances on Subatsk 2 are necessarily bounded by the performances on Subtask 1 (see also Table 1). Interestingly, the simple Exact matching approach provides a high performance, suggesting homogeneity between the training and the development sets. As intended,

Knowledge Sources	Subtask 1				Subtask 2		
	P	R	F1		P	R	F1
BETO (baseline)	90.8	87.7	89.2	Exact	93.21	82.66	87.62
				Levenshtein	92.50	84.69	**88.42**
BETO, UMLS	92.0	91.5	91.7	Exact	94.4	85.65	89.81
				Levenshtein	93.68	87.81	**90.65**
BETO, UMLS, POS	92.8	91.8	92.3	Exact	95.27	86.43	90.64
				Levenshtein	94.54	88.62	**91.48**
BETO, UMLS, POS, Group Gaz	92.9	92.9	**92.9**	Exact	95.38	87.39	91.21
				Levenshtein	94.62	89.60	**92.04**

Table 1
Development results

approximate string matching based on the Levenshtein distance improves recall att he cost of a small drop in precision; the overall F1 score, however, improves.

Error cases Using UMLS and BETO improves both precision and recall. Table 2 shows some error cases. Terms such as *piojos* (lice), *colonia* (a bird), *pulgas* (flee) are omissions (false negatives, FN) for BETO because these terms never occur in the training data. However, when adding a UMLS module to the system, they are correctly annotated (true positives, TP).

A special issue are disease mentions like *hepatitis*. While *hepatitis A* is a virus and should be annotated, hepatitis itself is a disease mention and does not refer to a living thing. The distinction is too subtle for the current system and a solution would involve more in depth language processing.

Entity	BETO	BETO, UMLS	BET, UMLS, POS
piojos	FN	TP	TP
pulgas	FN	TP	TP
colonia	FN	TP	TP
equipo de la UCI	FP	FP	TP
coronavirus 2019	FP	FP	FP
hepatitis	FP	FP	FP
enfermedades venéreas	FP	FP	FP
aeroalérgenos	FP	FP	TN
Transexual masculino	FN	FN	TP

Table 2
Comparison of error cases on development data

3.2. Competition phase

Table 3 shows the official competition results for our two submissions. The results show that our mi-RIM system performs well to the test dataset, suggesting robustness of the system and its integration of extant knowledge sources. Note that the success in Subtask 2 is dependent on

correct entity annotations in Subtask 1 and in fact the mean for both tasks is very close, as is the performance of our system, suggesting that Subtask 2 is generally well executed.

	Subtask 1				Subtask 2		
Knowledge Sources	μP	μR	μF1		μP	μR	μF1
BETO, UMLS, POS	93.79	92.51	93.15	Exact	96.40	86.92	91.42
				Levenshtein	94.90	89.06	**91.89**
BETO, UMLS, POS, Group Gaz	**93.85**	**92.56**	**93.20**	Exact	**96.41**	86.96	91.44
				Levenshtein	94.95	89.10	**91.93**
Competition mean	87.63	80.77	82.38		84.87	80.67	82.67
Competition STD	15.41	24.65	23.71		15.77	14.76	15.08

Table 3
Official competition results

4. Conclusion

This paper summarized the CLaC submission for the LivingNER challenge. We used the mi-RIM architecture to integrate heterogeneous knowledge sources such as BETO, UMLS, POS tags, and an ad-hoc gazetteer compiled on training data. mi-RIM accommodates external knowledge sources using independent but interacting modules. The results on both development and challenge data sets show that the modules interoperate well and performance is highest when all modules are used.

References

[1] A. Miranda-Escalada, E. Farré-Maduell, S. Lima-López, D. Estrada, L. Gascó, M. Krallinger, Mention detection, normalization classification of species, pathogens, humans and food in clinical documents: Overview of livingNER shared task and resources, Procesamiento del Lenguaje Natural (2022).

[2] O. Bodenreider, The Unified Medical Language System (UMLS): Integrating Biomedical Terminology, Nucleic acids research 32 (2004) D267–D270.

[3] P. Bagherzadeh, S. Bergler, Multi-input Recurrent Independent Mechanisms for leveraging knowledge sources: Case studies on sentiment analysis and health text mining, in: Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, 2021.

[4] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, GATE: A Framework and Graphical Development Environment for Robust NLP tools and applications, in: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02), 2002.

[5] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish pre-trained BERT model and evaluation data, in: PML4DC at ICLR 2020, 2020.

[6] R. Maldonado, M. Yetisgen, S. M. Harabagiu, Adversarial learning of knowledge embeddings for the unified medical language system, AMIA Summits on Translational Science Proceedings 2019 (2019) 543.
[7] D. Jurafsky, J. H. Martin, Speech and Language Processing, An Introduction to Natural Language Processing,Computational Linguistics, and Speech Recognition, 2020.
[8] P. Bagherzadeh, S. Bergler, Leveraging knowledge sources for detecting self-reports of particular health issues on social media, in: Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis, 2021.
[9] T. Mikolov, W.-t. Yih, G. Zweig, Linguistic regularities in continuous space word representations, in: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2013, pp. 746–751.
[10] R. Rehurek, P. Sojka, Software framework for topic modelling with large corpora, in: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, 2010.
[11] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in PyTorch, in: Neural Information Processing Systems (NIPS), 2017.
[12] D. P. Kingma, J. L. Ba, Adam: A method for stochastic optimization, in: Proceedings of the 3rd International Conference on Learning Representations, ICLR'15, 2015.