Dual Architecture for Name Entity Extraction and Relation Extraction with Applications in Medical Corpora

Ernesto Quevedo Caballero, Alejandro Rodriguez Perez, Tomas Cerny, Pablo Rivas
Baylor University
{ernesto_quevedo1,alejandro_rodriguez4,
tomas_cerny,pablo_rivas}
@baylor.edu

Abstract

There is a growing interest in automatic knowledge discovery in plain text documents. Automation enables the analysis of massive collections of information. Such efforts are relevant in the health domain which has a large volume of available resources to transform areas important for society when addressing various health research challenges. However, knowledge discovery is usually aided by annotated corpora, which are scarce resources in the literature. This work considers as a start point existent health-oriented Spanish dataset. In addition, it also creates an English variant using the same tagging system. Furthermore, we design and analyze two separated architectures for Entity Recognition and Relation Extraction that outperform previous works in the Spanish dataset. Finally, we evaluate their performance in the English version with such promising results.

1 Introduction

In recent decades, there has been significant growth in generating and collecting data in text form. This has caused a great interest of the scientific community in developing systems that assist the transformation of text into helpful knowledge. However, the sheer volume of information and the poorly unified semantic structure of documents written in natural language makes it difficult for researchers to efficiently find good results. The search for related research becomes much more complex when considering multiple languages. However, because Spanish is a less generalized language than English in terms of available computational resources, there are not many automatic information extraction systems available (Piad-Morffis et al., 2020).

The entity extraction and classification problem are formulated in the literature as Named Entity Recognition (NER) (Li et al., 2020). It is defined as the process of obtaining, from unstructured natural language text, a list of the sections that contain entities (Li et al., 2020). A related problem is Relation Extraction (RE) (Pawar et al., 2017), and classification is vent broader. It aims at determining which relations are established between the entities previously recognized in an input document (Pawar et al., 2017).

This paper improves on the models introduced by (Rodríguez-Péreza et al., 2020), obtaining two new separated architectures for the NER and RE problem, respectively. Next, it studies its performance in the Spanish dataset of the event eHealth-KD 2020 (Piad-Morffis et al., 2020) and an English dataset created by us based on the Spanish dataset, which showed that it outperforms the state of the art results in the Spanish dataset.

The paper is organized as follows. First, we present the datasets. Section 3 details both architectures for the NER and RE problems. Sections 4 performs experiments, and the last section concludes the paper.

2 Datasets

Datasets	(Train)	(Development)	(Testing)
Spanish	800	200	100
English	250	50	50

Table 1: Dataset distribution by number of sentences.

The dataset used was proposed in the event eHealth-KD in its 2020 edition (Piad-Morffis et al., 2020). Which contains tagged sentences with the entities and relations present in them. This event also is divided into two tasks. One task is for NER, and the second is for RE. We created the English version of this dataset based on the Spanish dataset's sentences translated to English and with adjusted relations.

3 Architectures

The system proposed in this paper solves both tasks separately and sequentially. Independent models are defined to solve NER and RE problems. The NER task takes the raw text of the input sentence and outputs two independent tag sequences: one in the BMEWO-V tag (Zavala et al., 2018) system for entity prediction (Rodríguez-Péreza et al., 2020), and another with tags corresponding to entity types (Concept, Action, Reference, Predicate) for classification purposes. The RE task is interpreted as a series of pairwise queries amongst the entities present in the target sentence.

Preprocessing: Given the target sentence and the highlighted entities input as raw text, some preprocessing is done to derive functional structures from such text. The input sentence must be tokenized first. Other preprocessing steps include character-level word decomposition, syntactic features extraction, and dependency parsing. To obtain a representation of the corresponding inputs, the models make use of BERT-based contextual embeddings, CNN-based character embeddings, POS-tag and dependency embeddings, and also embeddings for the BMEWO-V and Entity Type tag (Rodríguez-Péreza et al., 2020).

Data Augmentation: We implement a word replacement data augmentation algorithm (Dai and Adel, 2020) that automatically increase the dataset's size. This algorithm replaces entities words with the token `[MASK]`, and a pre-trained model of BERT is used to predict which word should replace the `[MASK]` token.

Entity Extraction Model: The model receives a sentence as a sequence of word vectors *S*. A distributed representation of each word is obtained concatenating contextual, character, and POS-tag embeddings, as described in the previous subsection. At a second level, the sequence of tokens is processed in both directions by a BiLSTM, resulting in a new sequence *P*. This sequence is processed by a stacked BiLSTM on top of the first one resulting in the sequence of vectors *P'*.

To assign tags in the BMEWO-V tag system to each word, and also a classification type (*Concept*, *Action*, *Reference*, *Predicate* and *None*), the next steps were split into two cases. To assign tags in the BMEWO-V tag system, the sequence *P'* is fed into a linear chain CRF layer that outputs the most likely tag sequence according to the Viterbi algorithm (Viterbi, 1967). In the second case, where a type must be assigned to each word, the sequence *P'* is fed into a Multiheaded Attention layer with eight heads, initialized with the value, key, and query vectors with the sequence *P'*. This layer will return a sequence vectors called *Z* which is fed to another CRF layer that outputs the most likely type sequence. The architecture for entity classification is shown in Figure 1

The first CRF layer produces a sequence of tags in the BMEWO-V tag system. A process is necessary to transform a tag sequence obtained from the CRF layer into a list of entities expected as output in Task A (Rodríguez-Péreza et al., 2020). An essential challenge in this process is that tokens belonging to an entity are not necessarily continuous in the sentence. Thus, the decoding process is divided into two stages. First, discontinuous entities are detected and then continuous entities. The set of tag sequences that must be interpreted as a group of discontinuous entities was narrowed to those that match the regular expressions:

$$(V+)((M*EO*)+)(M*E) \quad ((BO)+)(B)(V+)$$

The left regular expression corresponds to entities that share the initial tokens, and the right to those that share the final tokens. After detecting possible discontinuous entities, the second stage assumes that all the remaining entities appear as continuous sequences of tokens. Extracting the continuous entities is carried out as an iterative process over the tags sequence produced by the model using an automaton design by (Rodríguez-Péreza et al., 2020).

Relation Extraction Model: We design a Deep Learning model that takes as input the sentence and the two entities (e_1 and e_2) to classify its relationship. The model first encodes each of the structures S_{e_1}, S_{e_2} and $C(n_{e_1}, n_{e_2})$ in a vector. Where S_{e_1}, S_{e_2} are the subtree in the dependency parse tree (Liu et al., 2015) of the sentence of each entity respectively. $C(n_{e_1}, n_{e_2})$ is the path in the dependency tree from entity e_1 to entity e_2. A distributed representation of each word is obtained concatenating contextual, character, POS-tag, dependency, BMEWO-V and entity type embeddings.

A BiLSTM encodes the sequence of vectors in $C(n_{e_1}, n_{e_2})$. Then the output sequence of this BiLSTM, let's call it *P*, is fed into a Multiheaded Attention layer with five heads, initialized with the

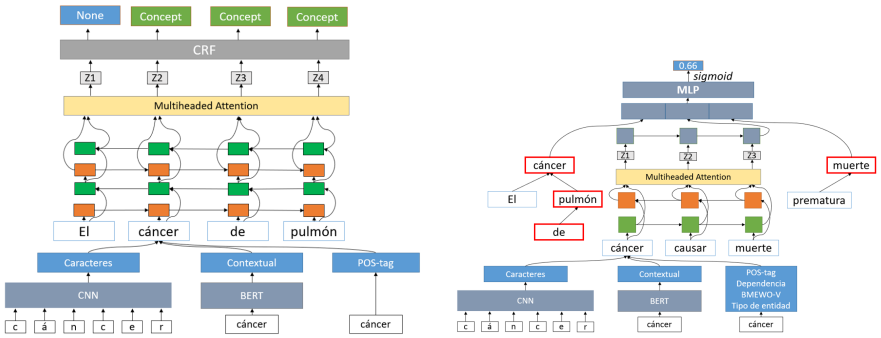

Figure 1: Left Entity Classification Model Architecture. Right Relation Extraction Model Architecture.

value, key, and query vectors with the sequence *P*. This layer returns a sequence of vectors called *Z* which is fed into a unidirectional LSTM to emphasize the direction of the potential relation. This results in a vector encoding the information present in $C(n_{e_1}, n_{e_2})$. A ChildSum Tree-LSTM (Tai et al., 2015) is applied independently over S_{e_1} and S_{e_2}. Vectors encoding the input structures are concatenated. The final output x is obtained by applying a sigmoid function to a linear transformation. If any component of the scoring output vector x exceeds a given threshold, then the relation with the maximum score is selected. Otherwise, no relation is reported. The threshold value is added as a hyperparameter and optimized using the development collection. Figure 1 shows the described architecture.

Teams	(A+B)	(A)	(B)	(A+B T)
Vicomtech	0.666	0.821	0.583	0.563
Our Approach (DA)	**0.633**	**0.829**	**0.637**	**0.587**
Our Approach	**0.631**	**0.828**	**0.637**	**0.561**
Talp-UPC	0.627	0.816	0.575	0.584
UH-MAJA-KD	**0.625**	**0.814**	**0.599**	**0.548**
IXA-NER-RE	0.558	0.692	0.633	0.479
baseline	0.395	0.542	0.131	0.138

Table 2: Results (measure F_1) in each scenario in the event *eHealth-KD* 2020. (DA) = data augmentation strategy.

4 Experiment and results

We evaluated the performance of the deep learning models in the Spanish language using the same testing dataset that in the competition eHealth-KD of 2020 (Piad-Morffis et al., 2020). Next, we evaluated the models performance with the English dataset using a testing set of 50 sentences but with the same metrics. The results are presented in F_1 measure with the respective definitions of precision and recall of the eHealth-KD of 2020 (Piad-Morffis et al., 2020; Piad-Morffis et al., 2020). Table 2 shows the results of other approaches in the same competition in comparison with our system.

System-Data-Augment	(A+B)	(A)	(B)	Size
Models with Spanish	0.633	0.829	0.637	1587
Models with English	0.572	0.781	0.550	1168

Table 3: Results on languages (measure F_1) from dataset evaluation using the same metric as in Table 2.

Table 2 shows the ranking in the Spanish dataset. Our approach obtains the best results in the NER task (A) and also in the RE task (B). Furthermore, our system simultaneously gets the best results in both tasks but in a general-purpose testing dataset (A + B T). However, a system is better in both tasks at the same time but in a medical-specific testing dataset (A + B). We believe the reason is the use of a joint model solving both tasks at the same time, instead of a model-specific for entities and others for relations (García-Pablos et al., 2020). Table 3 shows the best results after using the data augmentation algorithm from Section 3 in both datasets the Spanish and English.

5 Conclusions

We design two architectures for the NER and RE problems, with assessment in two datasets in the medical scope, showing that our models obtain better results than state-of-the-art work in the Spanish dataset. Finally, we introduce a new English dataset based on the health-oriented Spanish dataset of the eHealth-KD 2020. With such promising results we can obtain ontologies representations of the text using our models that can be applied in several other NLP tasks like Information Retrieval. Furthermore, we will be able to obtain the same ontology representation for Spanish and English text which can be used in Neural Machine Translation between these two languages.

References

Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. 2016. A full-text learning to rank dataset for medical information retrieval. In *European Conference on Information Retrieval*, pages 716–722. Springer.

Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. *arXiv preprint arXiv:2010.11683*.

Aitor García-Pablos, Naiara Perez, Montse Cuadros, and Elena Zotova. 2020. Vicomtech at ehealth-kd challenge 2020: Deep end-to-end model for entity and relation extraction in medical text. In *Proceedings of the Iberian Languages Evaluation Forum colocated with 36th Conference of the Spanish Society for Natural Language Processing, IberLEF@ SEPLN*, volume 2020.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*.

Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng Wang. 2015. A dependency-based neural network for relation classification. *arXiv preprint arXiv:1507.04646*.

Sachin Pawar, Girish K Palshikar, and Pushpak Bhattacharyya. 2017. Relation extraction: A survey. *arXiv preprint arXiv:1712.05191*.

Alejandro Piad-Morffis, Yoan Gutiérrez, Yudivian Almeida-Cruz, and Rafael Muñoz. 2020. A computational ecosystem to support ehealth knowledge discovery technologies in spanish. *Journal of biomedical informatics*, 109.

Alejandro Piad-Morffis, Yoan Gutiérrez, Suilan Estevez-Velarde, Yudivián Almeida-Cruz, Rafael Muñoz, and Andrés Montoyo. 2020. Overview of the ehealth knowledge discovery challenge at iberlef 2020. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020)*.

Filip Radlinski and Nick Craswell. 2010. Comparing the sensitivity of information retrieval metrics. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 667–674.

Alejandro Rodríguez-Péreza, Ernesto Quevedo-Caballeroa, Jorge Mederos-Alvaradoa, Rocío Cruz-Linaresa, and Juan Pablo Consuegra-Ayalaa. 2020. Uh-maja-kd at ehealth-kd challenge 2020: Deep learning models for knowledge discovery in spanish ehealth documents.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Andrew Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269.

Renzo M Rivera Zavala, Paloma Martínez, and Isabel Segura-Bedmar. 2018. A hybrid bi-lstm-crf model for knowledge recognition from ehealth documents. In *TASS@ SEPLN*, pages 65–70.

6 Appendix

6.1 Use Case Experiment

We present a use case experiment to evaluate the utility of the output of these two architectures in Information Retrieval systems. This experiment is done using the output of our system to build a graph ontology representation of the text, taking the entities as nodes and the relations as directed edges. We experimented with measuring the impact that this representation could have on Information Retrieval. For this, we used the Benchmark for Zero-shot Evaluation of Information Retrieval Models (BEIR) (Thakur et al., 2021), and we targeted the Reranking task in the health-oriented NFCorpus (Boteva et al., 2016). We design a score function with output between 0 and 1 to measure the relation of a query and a document.

The score function is based on the hypothesis that if we interpret the graph ontology as the knowledge representation of a text, then if a document is highly related to a query, the knowledge graph corresponding to the query should be a subgraph of the document's knowledge representation.

$$OScore(Q,D) = \frac{NScore(Q,D) + EScore(Q,D)}{2}$$

$$NScore(Q,D) = \frac{\sum_{v_i \in V_Q} NodeSim(v_i)}{2 * |V_Q|},$$

$$EScore(Q,D) = \frac{\sum_{e_i \in E_Q} EdgeSim(e_i)}{|E_Q|},$$

Definition 1 (Entity Similarity) *Given two ontology graphs $Q = (V_Q, E_Q)$ and $D = (V_D, E_D)$ and a pair of nodes $qnode \in V_Q$ and $dnode \in V_D$ the $EntSim(qnode, dnode)$ (Entity Similarity) is the cosine similarity of the **BERT** embeddings of the entities corresponding to each node.*

Definition 2 (Max Entity Related Node) *Given two ontology graphs $Q = (V_Q, E_Q)$ and $D = (V_D, E_D)$ and a node $qnode \in V_Q$. The Max Entity Related Node $dnode \in V_D$ to $qnode$ is the node with the highest value of $EntSim(qnode, dnode)$.*

Definition 3 (Node Similarity) *Given two ontology graphs $Q = (V_Q, E_Q)$ and $D = (V_D, E_D)$ the $NodeSim$ (Node Similarity) function of a node $qnode$ from Q is defined as finding its Max Entity Related Node $dnode$ in V_D. Then the value of $NodeSim$ is the value of $EntSim(qnode, dnode)$ increased by 1 if the classification of $qnode$ and $dnode$ as an entity is the same.*

Definition 4 (Edge Similarity) *Given two ontology graphs $Q = (V_Q, E_Q)$ and $D = (V_D, E_D)$, $e = (q_1, q_2) \in E_Q$, $q_1 \in V_Q$ and $q_2 \in V_Q$ and the Max Entity Related Node of q_1 and q_2 called as $d_1 \in V_D$ and $d_2 \in V_D$. The $EdgeSim$ (Edge Similarity) of e is 1 if exists the edge $e' = (d_1, d_2) \in E_D$ and it has the same label that e. $EdgeSim$ is 0 in any other case.*

Metric	**Ours**	**Combined**	**CEMMEB**
NDCG@1	0.2529	0.3846	0.4235
NDCG@10	0.2031	0.2564	0.2918
MAP@1	0.0228	0.0394	0.0465
MAP@10	0.0523	0.0785	0.0951
Recall@1	0.0228	0.0394	0.0465
Recall@10	0.0922	0.1103	0.1252
P@1	0.2529	0.3846	0.4235
P@10	0.1661	0.1903	0.2164

Table 4: Results of our score (**Ours**), the *cross-encoder/ms-marco-electra-base* (**CEMMEB**) used in the BEIR. The metrics reported are Normalized Discounted Cumulative Gain at k (NDCG@k), Mean Average Precision at k (MAP@k), Recall at k (Recall@k) and Precision at k (P@k) (Radlinski and Craswell, 2010; Thakur et al., 2021).

Table 4 shows the results in the Reranking task using our score function and an average of our score and the score obtained from one of the best-pretrained models that the framework offers for the Reranking task, which is *cross-encoder/ms-marco-electra-base* (Thakur et al., 2021). Even when the results of our score are the lowest in Table 4 we consider the results are not bad because we are using our models trained in the new English dataset that is still small, therefore, the performance of the models is low, especially the Relation Extraction model, which implies that the edge score will be weak. In our opinion, is that score the one more likely to give the improvement since the node score idea is in the most a relation score among words that are already contained in the original approach *cross-encoder/ms-marco-electra-base* (Thakur et al., 2021).