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

Anonymous

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 especially relevant in the health domain as advancements could use the 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 situation is particularly critical in the Spanish language, for which the volume of training resources is less widespread. 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 Extraction and Relation Recognition that outperform previous works in the Spanish dataset. With such promising results, we also evaluate their performance in the English version.

Introduction

In recent decades there has been a significant growth in the generation and collection of data in text form. This has caused a great interest of the scientific community in developing systems that assist the transformation of text into useful 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 find good results efficiently. In this domain is located the area of automatic information extraction in which, in turn, is present the problems of entity extraction and the relationships that are established between them.

The search for related research becomes much more complex when considering multiple languages. There are research areas where there are relevant results in more than one language, as is the case of medicine. We can find influencing results in English and also Spanish to give an example. 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 of that text that contain entities. Entities have been described in the literature differently, depending on the context, domain, and corpus used (Li et al. 2020). A related problem of Relation Extraction (RE) (Pawar, Palshikar, and Bhattacharyya 2017), and classification is vent broader. It aims at determining which relations are established between the entities previously recognized in an input document (Pawar, Palshikar, and Bhattacharyya 2017).

This paper improves on the models introduced by Rodríguez-Péreza et al. (2020), obtaining two new separated architectures for the Entity Extraction and Relation Recognition problem, respectively. Next, it studies its performance in the Spanish dataset of the event eHealth-KD 2020[1] and an English dataset created by us based on the Spanish dataset.

The paper is organized as follows. First, we present a section of related work. The next section elaborates on the datasets used and how the new English dataset was built. Then, Section 3 presents the design and details of both architectures for the NER and RE problems, respectively. Section 4 presents performed experiments. Finally, the last section concludes the paper and suggests futures work.

Background

The area of information extraction comprises problems that allow obtaining structured information from unstructured or semi-structured documents, usually using Natural Language Processing (NLP) techniques. Its general problem area includes NER, RE, and automatic ontology construction.

NER and RE are essential preprocessing steps for various problems such as Information Retrieval, Question Answering, Machine Translation, and others (Li et al. 2020). Several approaches have been found for NER in the literature like rules based (Zhang and Elhadad 2013), unsupervised learning (Nadeau and Sekine 2007), supervised based in features (Settles 2004; Li et al. 2020). In the last years, the most successful approaches have been found in deep learning techniches (Li et al. 2020). Successful deep learning ap-

[1] https://knowledge-learning.github.io/ehealthkd-2020/resources

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

Table 1: Distribution of both datasets, by number of sentences for traning, development and testing.

proaches are based on contextual encoders as Bidirectional Long Short Term Memory (BiLSTM), Convolutional Neural Networks (CNN), and Transformer architectures (Li et al. 2020). One final step in deep learning techniques for NER is the tag decodification stage, where the literature shows the use of Multilayer Perceptron with softmax activation, Recurrent Neural Networks (RNN) (Li et al. 2020) and Conditional Random Field (CRF) (Li et al. 2020; Lafferty, McCallum, and Pereira 2001).

RE also had its best results in the last year with deep learning approaches (Pawar, Palshikar, and Bhattacharyya 2017). Deep learning-based solutions to the RE problem are focused on sentence encoders using BiLSTM, CNN, and Transformers architectures (Pawar, Palshikar, and Bhattacharyya 2017). Also, deep learning solutions to the NER and RE problems need distributed representations of the input (Li et al. 2020; Pawar, Palshikar, and Bhattacharyya 2017). The most used representations in the last years are contextual embeddings of the word that can be obtained using pretrained Transformer models in large collections of text as **BERT** (Devlin et al. 2018), *word embeddings* (Mikolov et al. 2013) pretrained in large corpora or trained together with the model. In addition are used *character* embeddings that are trained with the model also, usually using **BiLSTM** or **CNN** based architectures (Li et al. 2020) and also Part of Speech tags (POS-tags) (Li et al. 2020). Particularly in RE, another highly used representation is the dependency tree associated with the sentence (Pawar, Palshikar, and Bhattacharyya 2017; Liu et al. 2015).

Research has also been done using the tagging system proposed in (Piad-Morffis et al. 2020). This tagging system is composed of four types of entities: *Concept, Action, Reference, Predicate* and a set of relations as *is-a, part-of, causes, has-property, entails, same-as*. Several models have been developed for the extraction and classification of entities and relations using this tagging system and a Spanish medical dataset in the event of eHealth-KD 2020 (Piad-Morffis et al. 2020).

From the output of these tasks (NER and RE), ontologies can be built and used in multiple problems as Information Retrieval (Asim et al. 2018). However, one limitation of the ontology learning supervised approaches is that it cannot always be used in numerous languages since there are not many datasets in more than one language that use the same tagging system (Piad-Morffis et al. 2020; Asim et al. 2018).

Datasets

The dataset used is the one proposed in the event eHealth-KD in its 2020 edition (Piad-Morffis et al. 2020). This dataset is composed of two collections of tagged sentences with the entities and relations present in them. The training collection is used to optimize the proposed models' parameters, and the development collection is used for the model selection. Finally, there is a testing collection to determine the final performance of the systems developed by the contestants. This event also is divided into two tasks. One task is for entity extraction and classification, and the second is for relation extraction and classification.

The English version of this dataset was created based on the Spanish dataset's sentences translated to English and with adjusted relations (additional dataset creation specifics are in the supplementary material). However, solely translating the dataset is not sufficient because the words used in English often express the same as in Spanish but do not mean the same in the full context, and the grammar is different. Therefore, entities change positions in the sentence, which implies that the relations have to be adjusted. Table 1 shows the distributions of both datasets.

Architectures

The Biarchitecture system proposed in this paper solves both tasks separately and sequentially. Thus, independent models were defined to solve NER and RE problems. The NER task is posed as a tag prediction problem that takes the raw text of the input sentence and outputs two independent tag sequences: one in the BMEWO-V tag (Zavala, Martínez, and Segura-Bedmar 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 tag None is included in the latter; consider the cases where no entity is present. Meanwhile, the RE task is interpreted as a series of pairwise queries amongst the entities present in the target sentence. A particular relation's existence is predicted upon features derived from both the sentence and the pair of entities.

Preprocessing Given the target sentence and the highlighted entities input as raw text, some preprocessing is done to derive functional structures from such text. Since both models make use of word-piece information, 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 the following features for each word:

Contextual embedding: BERT-based contextual embeddings with no further hyper tuning. Due to the BERT model's tokenization algorithm, a specific strategy is needed to merge words divided into multiple BERT tokens (e.g.the word cáncer might be divided [**cán, cer**]). In our case, it is done using the mean of the given vectors.

Character embeddings: CNN-based character embeddings. The input to such CNN is a sequence of alphabet indexes, those of the characters contained in the word.

POS-tag and Dependency embeddings: Embeddings intended to encode word-level syntactic features such as the POS-tag of the given the word and the dependency with its ancestor in the dependency parse tree.

BMEWO-V and Entity Type tags: BMEWO-V and entity type tags are used in the RE task and are obtained from Task A model outputs.

Named Entity Recognition Model

The model receives the 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 layer, resulting in two sequence vectors. The vectors on complementary positions of the two sequences are concatenated, resulting in a new sequence *P* with contextual-dependent vectors assigned to each token in the sentence. This sequence is looking to encapsulate semantic dependencies between the tokens of the sentence. The output sequence of the first BiLSTM is processed in both directions by a stacked BiLSTM on top of the first one, getting more representational power and resulting in the sequence of vectors *P'*:

$$P = \text{BiLSTM}(S) \tag{1}$$

$$P' = \text{StackedBiLSTM}(P). \tag{2}$$

The model has to assign tags in the BMEWO-V tag system to each word, and also a classification type in the classes *Concept*, *Action*, *Reference*, *Predicate* and *None*. To do so, the next steps were split into two cases. Both architectures are shown in the figures 1a and 1b

To assign tags in the BMEWO-V tag system to each word, 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). Let x_{tag} be the output corresponding to the BMEWO-V tag system and CRF_{tag} the CRF layer, then:

$$x_{tag} = \text{CRF}_{tag}(P'). \tag{3}$$

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 of attention vectors called *Z*, denoted as follows:

$$Z = \text{MultiHeadedAttention}(P', P', P'). \tag{4}$$

Finally, the sequence *Z* is also fed to another CRF layer that outputs the most likely type sequence. Let x_{type} be the output corresponding to the entity type and CRF_{type} the linear chain CRF layer, then:

$$x_{type} = \text{CRF}_{type}(Z). \tag{5}$$

The first CRF layer produces a sequence of tags in the BMEWO-V tag system. Table 2 shows the description of the 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). This process from now on will be referred to as decoding. 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, at a second moment, continuous entities.

Tag	Meaning
B	Beginning of an entity
M	Middle of an entity
E	End of an entity
W	Single-token entity
V	Two or more entities overlap in that token
O	Token does not represent anything

Table 2: BMEWO-V tag system meaning.

The set of tag sequences that must be interpreted as a group of discontinuous entities were narrowed to those that match the regular expressions:

$$(V+)((M*EO*)+)(M*E) \tag{6}$$

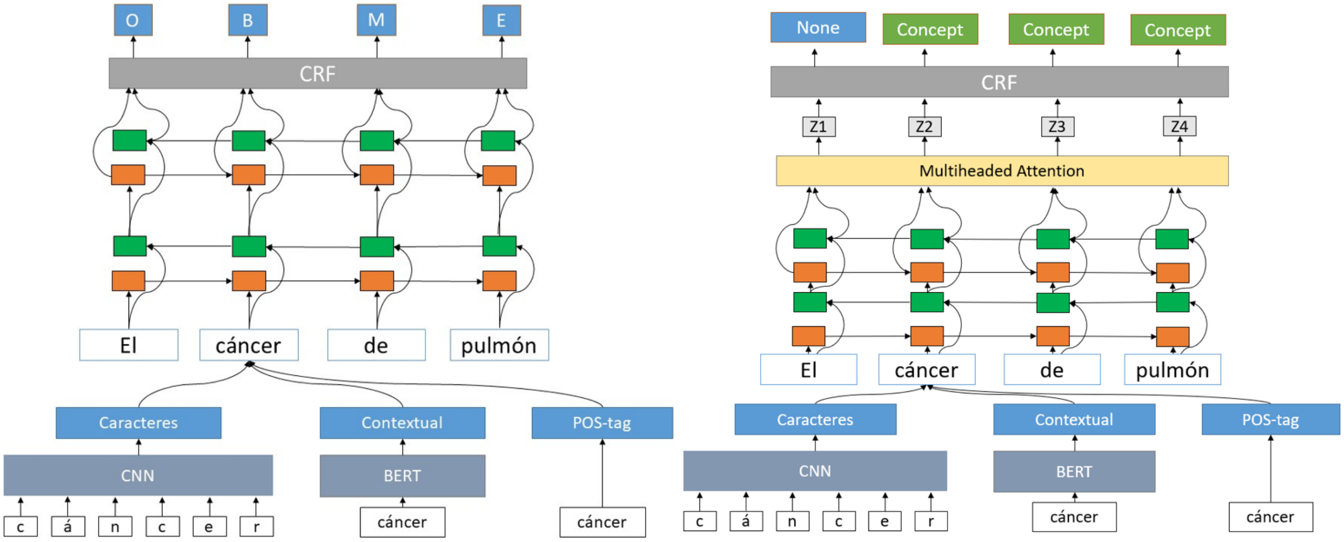
$$\text{and } ((BO)+)(B)(V+). \tag{7}$$

The former 6 corresponds to entities that share the initial tokens, and the latter 7 to those that share the final tokens. These two capture most of the desired discontinuous entities. Among the examples of the former case, it is found the fragment ***cáncer de pulmón y de mama***, tagged as V-M-E-O-M-E, where entities *cáncer de pulmón* and *cáncer de mama* are found. And, as example of the latter, the fragment ***tejidos y órganos humanos***, tagged as B-O-B-V, where entities *tejidos humanos* and *órganos humanos* are found. When a match is detected and the entities are extracted, then all the tags in that fragment are set to the tag O.

After detecting possible discontinuous entities, the second stage begins assuming 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. Due to limitations in the BMEWO-V system, the procedure also assumes that the maximum overlapping depth is 2. Assuming otherwise only makes the process ambiguous and does not capture much more information since a deeper overlapping is not frequent on the training and development collections. Given this, at most, two partially-constructed entities are maintained across the procedure. In each iteration, these two entities are created, extended with new tokens, or reported as completed, following rules defined considering only the previous and the current tag.

According to evaluations performed in the training and development collections, the process of decoding correctly labeled sequences extracts more than 98% of the entities present in the Spanish dataset.

After identifying the entities, we classify each of them according to its type, using a voting system based on the second CRF layer's output. The system had previously assigned to each word in the input sentence, one of the entity types, in this case one between: `Concept`, `Action`, `Predicate` or `Reference`. Each word produces a vote for each entity it belongs to, according to the assigned type. Then, each



(a) Entity Extraction Model Architecture for BMEWO-V tags.

(b) Entity Extraction Model Architecture for Classification.

Figure 1: On the left the Entity Extraction Model Architecture for BMEWO-V tags. On the right the Entity Extraction Model Architecture for Classification.

entity is classified according to the type that obtained the highest number of votes. If the voting is even, `Concept` is assumed since it is the most frequent by a wide margin in the collections studied.

Relation Extraction Model

The complete information to solve the RE task is found in the whole input sentence. However, some authors claim that the dependency tree associated with the input sentence condenses the essential pieces of information and discards the misleading ones (Liu et al. 2015; Xu et al. 2015). Aiming at determining a possible relation between two entities, the system presented uses input structures derived from the dependency parse tree associated with the target sentence to obtain information from the sentence and the entity pair.

One of the criteria taken into consideration to establish a dependency relationship with a header H in a syntactic construction C, is the fact that H could replace C (Zwicky 1985). Moreover, H could semantically determine C. On the other hand, multiple-word entities often occur entirely in a dependency subtree rooted at one of its tokens. Given a sentence and its dependency tree T, we define such subtree of T corresponding to an entity e, as **relevant tree for *e***, and it is denoted further on as S_e. The root is called the **core of the entity *e***, and it is denoted n_e.

Another important definition, vastly used in literature to address this task, the is **dependency path between two tokens** t_1 and t_2. From now on, it will be referred to as $C(t_1, t_2)$. The before-mentioned structures are fed into a Deep Neural Network that outputs a vector whose length is the same as the relations set. Each component of such vector is independent of each other and measures how certain is the model that the respective relation between the input entities appears.

To do so, 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. Either S_{e_1} and S_{e_2} or $C(n_{e_1}, n_{e_2})$ are formed by words from the input sentence. A distributed representation of each word is obtained concatenating contextual, character, POS-tag, dependency, BMEWO-V and entity type embeddings, as described in the previous subsection.

To compute the output vector, a BiLSTM layer encodes the sequence of vectors associated to the words in $C(n_{e_1}, n_{e_2})$ to include bidirectional information in the representation:

$$P = \text{BiLSTM}(C(n_{e_1}, n_{e_2})). \tag{8}$$

Then the sequence *P* is fed into a Multiheaded Attention layer with five heads, initialized with the value, key, and query vectors with the sequence *P*. This layer returns a sequence of attention vectors called *Z*, defined as follows:

$$Z = \text{MultiHeadedAttention}(P, P, P). \tag{9}$$

This output is fed into a unidirectional LSTM layer to emphasize the direction of the potential relation, processing the sequence Z from the origin to the destination. This results in a vector p encoding the information present in $C(n_{e_1}, n_{e_2})$:

$$p = \text{LSTM}(Z). \tag{10}$$

At the same time, a ChildSum Tree-LSTM (Tai, Socher, and Manning 2015) is applied independently over S_{e_1} and S_{e_2} (i.e the representations are obtained separately but using the same set of weights):

$$t_{e_1} = \text{TreeLSTM}(S_{e_1}) \tag{11}$$

$$t_{e_2} = \text{TreeLSTM}(S_{e_2}) \tag{12}$$

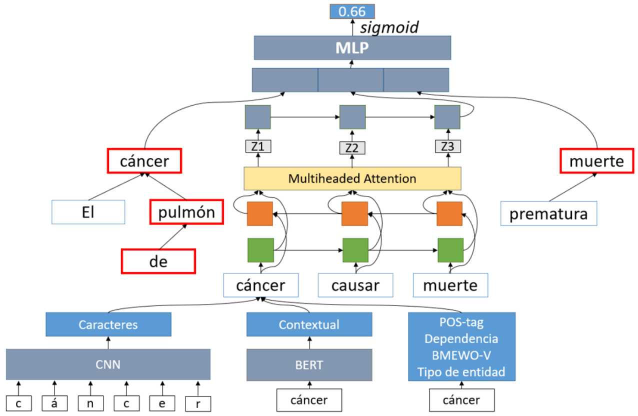


Figure 2: Relation Extraction Model Architecture.

Vectors encoding the input structures are concatenated. The final output x is obtained by applying a sigmoid function to a linear transformation of it as follows:

$$r = [t_{e_1}; t_{e_2}; p] \tag{13}$$

$$x = \sigma(W^{(x)} r + b^{(x)}) \tag{14}$$

According to the scores present in the output vector x, if any of its components exceeds a given threshold, then the relation with the maximum score is said to exist. If none of the scores is greater than such threshold, then no relation is reported. The threshold value is added as a hyperparameter and optimized using the development collection. Notice that this approach allows us to disregard the use of a fake relation `none`. Figure 2 shows the described architecture.

Parameters Setup and Training

For both models, the training procedure was carried out using only the training collection.

Since the CRF layer is intended to maximize the probability of obtaining a desired tag sequence y given an input feature vector X, the Task A model is trained to minimize the negative log of the probability $P(y|X)$. Let U and T be the CRF emissions and transition matrixes, respectively. Then, that probability is defined as the normalized exponential:

$$P(y|X) = \frac{\exp\left(\sum_{k=1}^{l} U(x_k, y_k) + \sum_{k=1}^{l-1} T(y_k, y_{k+1})\right)}{Z(X)},$$

where Z is a normalization factor depending on the input vector X. And the loss function is defined in terms of X and y as follows:

$$\ell(X, y) = -\log(P(y|X)).$$

In the case of Task B model, since each output component is independent to each other, the model is trained to minimize a binary cross-entropy function over the output vector. Let k be the number of relations, x the output vector and y the target vector, the loss function is computed as follows:

$$\ell(x, y) = \frac{1}{k} \sum_{1 \leq i \leq k} [y_i \cdot \log x_i + (1 - y_i) \cdot \log(1 - x_i)].$$

Parameter	Value	Parameter	Value
Input embeddings size			
Contextual†	3072	Contextual*	768
Character	50	Character	50
POS-tag	50	POS-tag	50
		Dependency	50
		BMEWO-V tags	50
		Entity type	50
Neural network			
CNN hid. sz.	100	CNN hid. sz.	100
2D Dropout	0.5	BiLSTM h. sz.	100
BiLSTM_1 h.sz.	300	Dropout rate	0.2
Dropout_1 rate	0.5	LSTM hid. sz.	50
BiLSTM_2 h. sz.	300	Dropout rate	0.5
Multihead att hds	8	Multihead att hds	5
Dropout_2 rate	0.5	Tree-LSTM sz.	50
		Dropout rate	0.5
Training			
Optimizer	Adam	Optimizer	Adam
Learning rate	0.001	Learning rate	0.001
Epochs	50	Epochs	30
Total params 9,935,538		**Total params** 5,947,763	

Table 3: Hyperparameter setup for NER (left) and RE (right) models. Annotations: †last four, *last layer.

As explained before, the model output does not make use of the fake `none` relation. A negative sampling strategy is used to optimize the model with examples where no relation is present. A negative sample is nothing more than a training example where the target output is the null vector. Such sampling is performed using a fixed proportion of unrelated entity pairs.

Dropout strategies were used during the training procedure in both models to reduce overfitting. For Task A, two dropouts layers were stacked after the first and the second BiLSTM, and a spatial dropout 2D was added after the CNN layer was used to compute the character embedding of words. In the Task B model, three dropout layers were stacked after BiLSTM, LSTM, and TreeLSTM layers, respectively.

The number of epochs was selected empirically, based on the convergence of the models, as learning curves showed. We carried cross-validation for hyperparameter tuning and model selection using the development collection. Table 3 shows the hyperparameter setup for both models.

Data Augmentation

Also, the implementation of a word replacement data augmentation algorithm (Dai and Adel 2020) will automatically increase the dataset's size. This algorithm first goes for each sentence in the dataset and searches for an entity composed of only one word. Then it changes that word with the token

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 4: Results (measure F_1) obtained from the evaluation of the systems in the Spanish dataset provided in the event *eHealth-KD* 2020 and the newly created English dataset. In both datasets a data augmentation strategy was used. The size column shows the size in sentences of the augmented dataset. Scenario A is only entities, B is only relations, and (A+B) is the result using both at the same time.

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
UH-MatCom	0.557	0.795	0.545	0.373
SINAI	0.421	0.825	0.462	0.281
HAPLAP	0.395	0.542	0.316	0.138
baseline	0.395	0.542	0.131	0.138
ExSim	0.246	0.314	0.131	0.122

Table 5: Results (measure F_1) in each scenario of the competition, sorted by scenario 1 in the event *eHealth-KD* 2020. The (A + B T) scenario is both tasks together but in an evaluation dataset of general purpose. The results are obtained from the evaluation of the systems of each team in the dataset provided in the event. The system using the models of this work and the previous version of these models are highlighted in black. The label (DA) means our approach using the data augmentation strategy.

`[MASK]`, and a pre-trained model of BERT is used to predict which word should replace the `[MASK]` token. If the predicted word is different from the previous word, then a new sentence is created using this new predicted word. The idea is that there is a high probability that the word predicted will also be an entity and will have the same classification. Still, there exists the possibility of errors. One of those cases is when words like prepositions or pronouns are used too much. Therefore, to decrease the likelihood of an entity being replaced by a pronoun, preposition, or another non-entity word, we restrict that the predicted word cannot be a stopword. Two strategies were implemented. The first strategy is to add a new sentence for each word replaced. This means that for only one sentence, more than one new sentence can be generated, the number of new sentences will be bounded by the number of words that fit the criteria to be replaced. The second strategy is to add a new sentence for each existing sentence by changing all the possible words in the already existing sentence. This means that it will double the size of the dataset at most, but new sentences will be varied.

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 model training with the English dataset using a testing set of 50 sentences but with the same metrics. Also, Table 5 shows the results of the other approaches in the same competition in the Spanish language in comparison with our approach. 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).

As can be seen in the Spanish dataset results in Table 5, our approach obtains the best results in the task of only extracting and classifying entities (A) and also in the task of only extracting and classifying the relations (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). Obtaining functions that jointly optimize both tasks have a great complexity (García-Pablos et al. 2020). However, the fact that our proposal shows competitive results allows us to suppose that training separate models to solve the two tasks is still a promising line of research.

Table 4 shows the best results after using the data augmentation algorithm proposed in Section . The strategy of a new sentence for each word changed worked better for the English dataset since its original size is still too small. However, this strategy brings more noise and bias to a bigger dataset like the Spanish one. For that reason, we use the strategy of a new sentence in the Spanish dataset to change all the possible words in an already existing sentence. We also believe that the use of this data augmentation strategy is one of the main reasons for the improvement of the results in the task (A + B T). Since that, the new words added by the pre-trained model of BERT **bert-base-multilingual-cased** during the prediction are general-purpose and not medical-specific. Also, from the results in Table 4 can be seen that the English dataset is still small since the performance of the models trained on it is low.

Conclusions

This work designs two separated architectures for the NER and RE problems and assesses them in both datasets, showing that our models obtain great results compared to 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 using the same tagging system, allowing future work from a multilingual approach using both datasets. We intend to continue increasing the size of the English dataset, and improve the performance of the models.

References

Asim, M. N.; Wasim, M.; Khan, M. U. G.; Mahmood, W.; and Abbasi, H. M. 2018. A survey of ontology learning techniques and applications. *Database*, 2018.

Dai, X.; and Adel, H. 2020. An analysis of simple data augmentation for named entity recognition. *arXiv preprint arXiv:2010.11683*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

García-Pablos, A.; Perez, N.; Cuadros, M.; and Zotova, E. 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 co-located with 36th Conference of the Spanish Society for Natural Language Processing, IberLEF@ SEPLN*, volume 2020.

Lafferty, J. D.; McCallum, A.; and Pereira, F. C. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, 282–289.

Li, J.; Sun, A.; Han, J.; and Li, C. 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*.

Liu, Y.; Wei, F.; Li, S.; Ji, H.; Zhou, M.; and Wang, H. 2015. A dependency-based neural network for relation classification. *arXiv preprint arXiv:1507.04646*.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Nadeau, D.; and Sekine, S. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1): 3–26.

Pawar, S.; Palshikar, G. K.; and Bhattacharyya, P. 2017. Relation extraction: A survey. *arXiv preprint arXiv:1712.05191*.

Piad-Morffis, A.; Gutiérrez, Y.; Almeida-Cruz, Y.; and Muñoz, R. 2020. A computational ecosystem to support eHealth Knowledge Discovery technologies in Spanish. *Journal of biomedical informatics*, 109: 103517.

Piad-Morffis, A.; Gutiérrez, Y.; Estevez-Velarde, S.; Almeida-Cruz, Y.; Muñoz, R.; and Montoyo, A. 2020. Overview of the eHealth Knowledge Discovery Challenge at IberLEF 2020. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020)*.

Rodríguez-Péreza, A.; Quevedo-Caballeroa, E.; Mederos-Alvaradoa, J.; Cruz-Linaresa, R.; and Consuegra-Ayalaa, J. P. 2020. UH-MAJA-KD at eHealth-KD Challenge 2020: Deep Learning Models for Knowledge Discovery in Spanish eHealth Documents. *Proceedings of the Iberian Languages Evaluation Forum*.

Settles, B. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications (NLP-BA/BioNLP)*, 107–110.

Tai, K. S.; Socher, R.; and Manning, C. D. 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)*, 1556–1566.

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

Xu, Y.; Mou, L.; Li, G.; Chen, Y.; Peng, H.; and Jin, Z. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, 1785–1794.

Zavala, R. M. R.; Martínez, P.; and Segura-Bedmar, I. 2018. A Hybrid Bi-LSTM-CRF model for Knowledge Recognition from eHealth documents. In *TASS@ SEPLN*, 65–70.

Zhang, S.; and Elhadad, N. 2013. Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. *Journal of biomedical informatics*, 46(6): 1088–1098.

Zwicky, A. M. 1985. Heads1. *Journal of linguistics*, 21(1): 1–29.