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

Anonymous ACL submission

## 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 uses a health-oriented Spanish dataset, and 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. Finally, we perform a use case experiment to evaluate the utility of the output of these two architectures in Information Retrieval systems.

## 1 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 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 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. Finally, a use case experiment is designed using the Benchmark for Zero-shot Evaluation of Information Retrieval Models (BEIR) (Thakur et al., 2021) to show the impact of the graph ontologies built from the output of both architectures in Information Retrieval Systems. In addition, we defined a score function to assess the similarity between two texts based on their ontological representation.

The paper is organized as follows. First, we present a section of related work. The next section

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

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. The consequent Section 5 shows the use case experiment, the new similarity score function defined, and the results. Finally, the last section concludes the paper and suggests futures work.

## 2 Background

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 approaches 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 et al., 2001).

RE also had its best results in the last year with deep learning approaches (Pawar et al., 2017). Deep learning-based solutions to the RE problem are focused on sentence encoders using BiLSTM, CNN, and Transformers architectures (Pawar et al., 2017). Also, deep learning solutions to the NER and RE problems need distributed representations of the input (Li et al., 2020; Pawar et al., 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 et al., 2017; Liu et al., 2015).

| 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.

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).

## 3 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. 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.

## 4 Architectures

The Dual Architecture 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 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 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.

### 4.0.1 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.

**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 NER model outputs.

## 4.1 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'*:

$$P = \text{BiLSTM}(S), \quad P' = \text{StackedBiLSTM}(P).$$

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:

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

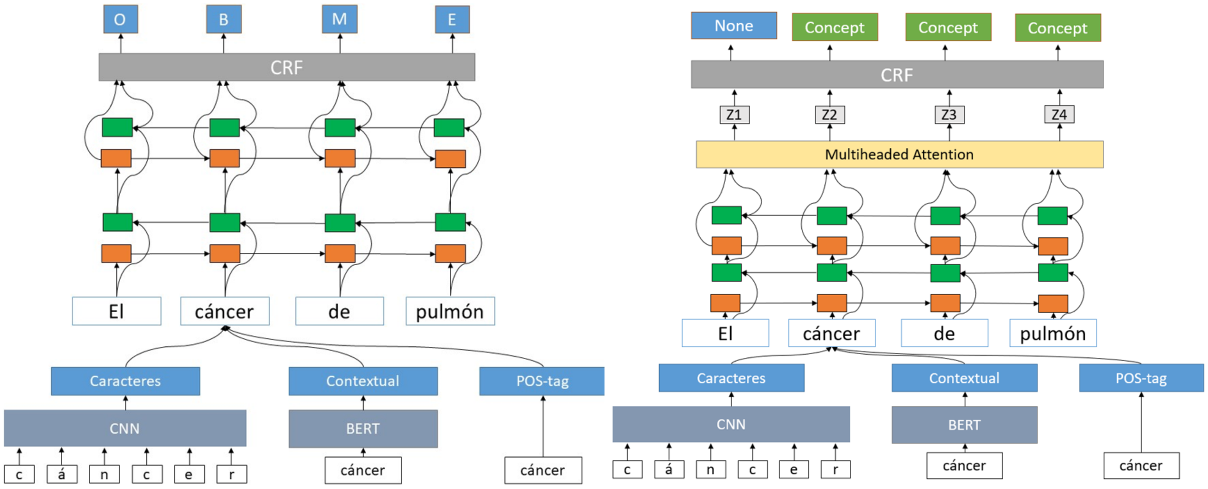
In the second case, where a type must be assigned to each word, the sequence *P'* is fed into a Multiheaded Attention layer with eight heads, initialized with the value, key, and query vectors with the sequence *P'*. This layer will return a sequence of attention vectors called *Z*, denoted as follows:

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

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:

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

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



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

(b) Entity Extraction 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.

| 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.

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.

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

$$(V+)((M*EO*)+)(M*E) \qquad (1)$$

$$\text{and } ((BO)+)(B)(V+). \qquad (2)$$

The former 1 corresponds to entities that share the initial tokens, and the latter 2 to those that share the final tokens. These two capture most of the desired discontinuous entities. 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. 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 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.

### 4.2 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:

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

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:

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

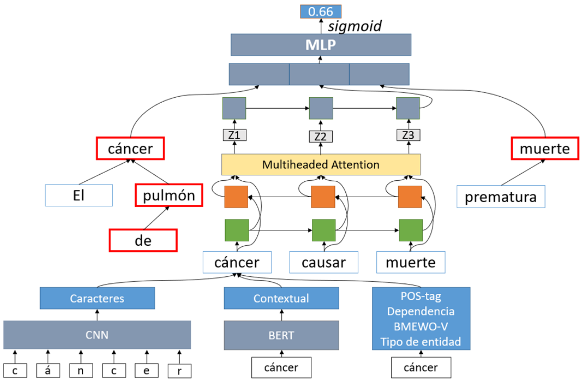


Figure 2: Relation Extraction Model Architecture.

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 $C(n_{e_1}, n_{e_2})$:

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

At the same time, a ChildSum Tree-LSTM (Tai et al., 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):

$$t_{e_1} = \text{TreeLSTM}(S_{e_1}), \quad t_{e_2} = \text{TreeLSTM}(S_{e_2})$$

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

$$r = [t_{e_1}; t_{e_2}; p], \quad x = \sigma(W^{(x)} r + b^{(x)})$$

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.

### 4.3 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

| **Parameter** | **Value** | **Parameter** | **Value** |
|---|---|---|---|
| **Input Embeddings size** | | | |
| Contextual$^{\dagger}$ | 3072 | Contextual$^{\star}$ | 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: $^{\dagger}$last four, $^{\star}$last layer.

probability is defined as the normalized exponential:

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

where $Z$ is a normalization factor depending on the input vector $X$. 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 $\ell(x,y)$ is computed as follows:

$$\frac{1}{k}\sum_{1\leq i\leq k} y_i \cdot \log x_i + (1-y_i)\cdot \log(1-x_i). \quad (3)$$

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. For hyperparameter tuning and model selection, a cross-validation process was carried out using the development collection. Table 3 shows the hyperparameter setup for both models.

### 4.4 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 `[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. 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 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 the new sentences will be more different than the existent ones.

## 5 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 4 shows the results of the other approaches in the same competition in

| Teams | (A+B) | (A) | (B) | (A+B T) |
|---|---|---|---|---|
| Vicomtech | 0.666 | 0.821 | 0.583 | 0.563 |
| **Ours (DA)** | **0.633** | **0.829** | **0.637** | **0.587** |
| **Ours** | **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 4: 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 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.

| 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 5: 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.

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). Also, an overview of the different models presented in Table 4 can be found in (Piad-Morffis et al., 2020).

As can be seen in the Spanish dataset results in Table 4, 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 5 shows the best results after using the data augmentation algorithm proposed in Section 4. 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.

## 6 Use Case Experiment

The output of this dual architecture system can be used to build a graph ontology representation of the text, taking the entities as nodes and the relations as directed edges. We built an experiment to measure 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). To address this task in the framework, a score function with an output between 0 and 1 has to be used to measure how related are a query and a document.

We defined a score function 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. The following equations and definitions detail the score function that we called Ontology Score:

$$OScore(Q,D) = \frac{NScore(Q,D) + EScore(Q,D)}{2}, \quad (4)$$

$$NScore(Q,D) = \frac{\sum_{v_i \in V_Q} NodeSim(v_i)}{2 * |V_Q|}, \quad (5)$$

$$EScore(Q,D) = \frac{\sum_{e_i \in E_Q} EdgeSim(e_i)}{|E_Q|}, \quad (6)$$

where $Q = (V_Q, E_Q)$ and $D = (V_D, E_D)$ represent the graph ontology obtained from the query and document, respectively.

**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*[2] *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.*

Table 6 shows the results in the Reranking task using our score function and also 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 6 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 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).

| Metric | **Ours** | **Combined** | **CEMMEB** |
|---|---|---|---|
| NDCG@1 | 0.2529 | 0.3846 | 0.4235 |
| NDCG@10 | 0.2031 | 0.2564 | 0.2918 |
| NDCG@100 | 0.1479 | 0.1740 | 0.1877 |
| MAP@1 | 0.0228 | 0.0394 | 0.0465 |
| MAP@10 | 0.0523 | 0.0785 | 0.0951 |
| MAP@100 | 0.0664 | 0.0898 | 0.1037 |
| 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 6: Results of our score (**Ours**), the *cross-encoder/ms-marco-electra-base* (**CEMMEB**) used in the BEIR framework, and the combination of both by taking the mean. 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).

## 7 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. Also, a score similarity function was presented for two ontology graphs and a use case experiment using the BEIR framework and the NFCorpus to evaluate the output of both models after building a graph ontology from the text using our architectures. 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, improve the performance of the models and the score similarity function of two ontologies and evaluate in more datasets that BEIR offers besides NFCorpus.

[2] We use the pretrained model **bert-large-cased** to get the embeddings.

## 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*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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 co-located with 36th Conference of the Spanish Society for Natural Language Processing, IberLEF@ SEPLN*, volume 2020.

John D Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289.

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*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

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

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*, 109:103517.

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. *Proceedings of the Iberian Languages Evaluation Forum*.

Burr Settles. 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 (NLPBA/BioNLP)*, pages 107–110.

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.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 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*, pages 1785–1794.

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.

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

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