

UH-MAJA-KD at eHealth-KD Challenge 2020: Deep Learning Models for Knowledge Discovery in Spanish eHealth Documents

Alejandro Rodríguez-Pérez^a, Ernesto Quevedo-Caballero^a, Jorge Mederos-Alvarado^a, Rocío Cruz-Linares^a and Juan Pablo Consuegra-Ayala^a

^aFaculty of Math and Computer Science, University of Habana, 10200 La Habana, Cuba

Abstract

This paper describes the solution presented by the UH-MAJA-KD team at IberLEF 2020: eHealth Knowledge Discovery Challenge. Separate strategies were developed to solve Tasks A and B, both based on Deep Learning models using contextual embeddings obtained from a pretrained BERT model, and some other syntactic features. We propose a strategy using a hybrid model for Task A that uses Stacked Bidirectional LSTM layers as contextual encoders, and linear chain Conditional Random Fields as tag decoders. The system addresses Task B in a pairwise-query fashion, encoding information about the sentence and the given pair of entities using syntactic structures derived from the dependency parse tree, by the means of LSTM-based Recurrent Neural Networks. The output is obtained scoring every possible relation via a Multilayer Perceptron with a sigmoid activation function. Our model was able to get a high performance in all four tasks of the competition. The system was ranked third in the main evaluation scenario, with a 0.001 difference with the second place. Additionally, it was ranked second in the evaluation responsible for measuring the performance in Task B, considered the hardest one in previous editions of the challenge.

Keywords

Knowledge Discovery, Information Extraction, Named Entity Recognition, Relation Extraction, Deep Learning

1. Introduction

The IberLEF 2020: eHealth Knowledge Discovery Challenge [1] (or eHealth-KD 2020 for simplicity) is an event that aims to push the state of the art boundary in the knowledge discovery area, particularly for medical-content text. It counts so far two previous editions [2, 3], defining similar tasks. The challenge is divided into two tasks: A and B; one for entity extraction and classification, and the other oriented to the extraction of semantic relationships between pairs of such entities.

This paper describes the solution presented by the UH-MAJA-KD team at eHealth-KD 2020. It proposes a hybrid model for Task A that uses Stacked Bidirectional Long Short Term Mem-


Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020)

EMAIL: a.rodriguez4@estudiantes.matcom.uh.cu (A. Rodríguez-Pérez); e.quevedo@estudiantes.matcom.uh.cu (E. Quevedo-Caballero); jorge.mederos@estudiantes.matcom.uh.cu (J. Mederos-Alvarado); rociocl@matcom.uh.cu (R. Cruz-Linares); jpconsuegra@matcom.uh.cu (J.P. Consuegra-Ayala)

ORCID: 0000-0002-0069-8950 (R. Cruz-Linares); 0000-0003-2009-393X (J.P. Consuegra-Ayala)



© 2020 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

ory (BiLSTM) layers as contextual encoders, because of the sequential structure of the input and its widely use in the literature [4] for addressing the Named Entity Recognition (NER) problem. Also, a linear chain Conditional Random Field (CRF) [5] is used as the tag decoder architecture for the model, for it has been used in many Deep Learning based NER systems on top of a BiLSTM, with successful results [4]. The system addresses the Relation Extraction (RE) task in a pairwise-query fashion, encoding information about the sentence and the given pair of entities using syntactic structures derived from the dependency parse tree, and by the means of Long-Short Term Memory (LSTM) based Recurrent Neural Networks (RNN) to achieve such purpose. Dependency information has proved useful in solving RE task in various benchmark datasets [6, 7, 8]. The output is obtained scoring every possible relation via a Multilayer Perceptron (MLP) with a sigmoid activation function.

The rest of the paper is organized as follows. Section 2 explains in detail the proposed system. The results of the model in the several scenarios evaluated during the eHealth-KD 2020 event are presented in Section 3. Section 4 analyses briefly matters of interest related to the development and performance of the models. Finally, the conclusions of the work are shown in Section 5.

2. System Description

The system proposed 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 system for entity prediction [9], and another with tags corresponding to entity types (Concept, Action, Reference, Predicate) for classification purposes. The tag None is included in the latter for 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. Hence, it predicts the existence of a certain relation upon features derived from both the sentence and the pair of entities.

2.1. Preprocessing

Given the target sentence and the highlighted entities input as raw text, some preprocessing is done in order to derive useful 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 hypertuning.

Due to BERT model's tokenization algorithm, a certain strategy is needed to merge words divided into multiple BERT tokens (e.g, word **cáncer** might be divided in [**cán**, **cer**]). In our case, it is done using the mean of the given vectors. Each model uses the concatenation of a number of BERT output 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 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 RE task and are obtained from Task A model outputs.

Contextual embeddings are pretrained with no further hypertuning; whereas the remaining embeddings' weights are optimized when training the corresponding model.

2.2. Named Entity Recognition Model

The model receives the sentence as a sequence of words 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' .

$$P = BiLSTM(S) \quad P' = StackedBiLSTM(P)$$

Sequence P' is fed into two linear chain CRF layers, that output the most likely tag sequences according to the *Viterbi* algorithm [10]. Let x_{tag} and x_{type} be the outputs corresponding to the BMEWO-V tag system and the entity type, respectively; and CRF_{tag} and CRF_{type} the respective linear chain CRF layers, then:

$$x_{tag} = CRF_{tag}(P') \quad x_{type} = CRF_{type}(P')$$

Figure 1 shows the described architecture.

The first CRF layer produces a sequence of tags in the BMEWO-V tag system. This classification corresponds to B for the begin of an entity, M for tokens in the middle, E for the ending token, W in the case of tokens that are an entity themselves and O for tokens that do not represent anything. It also takes into account the possibility of overlapping entities, including the tag V in such cases.

A process is necessary to transform the tag sequences got from the CRF layers into a list of entities expected as output for Task A [9]. This process from now on will be referred to as **decoding**. There is an important challenge in this process: tokens belonging to an entity are not necessarily continuous in the sentence. Taking this into account, the decoding process is

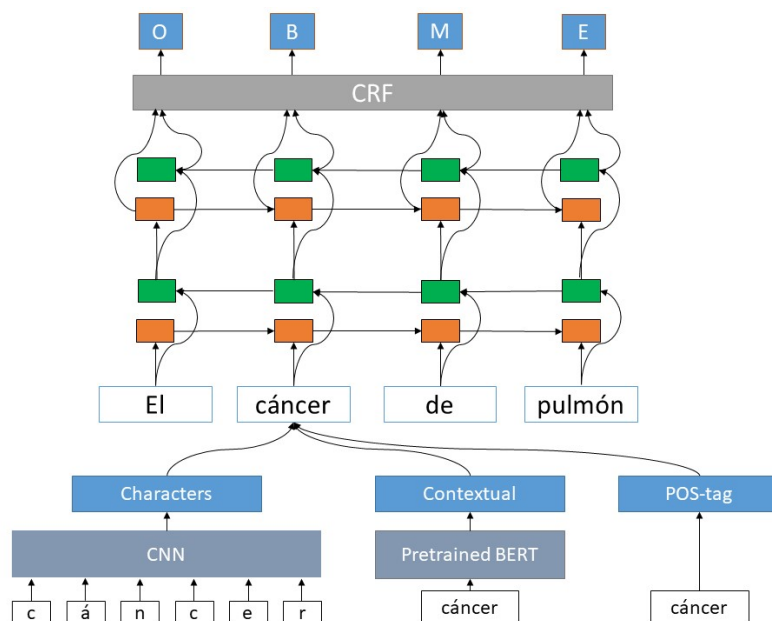


Figure 1: Task A model architecture (only showing one output).

divided into two stages. First, discontinuous entities are detected and then, at a second moment, continuous entities.

In accordance to Spanish correct use, the set of tag sequences that must be interpreted as a group of discontinuous entities were reduced to those that match the regular expressions $(V+)((M^*EO^*)+)(M^*E)$ and $((BO^+) +)(B)(V^+)$. The former corresponds to entities that share their initial tokens, and the latter to those that share their 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, all the tags in that fragment are set to tag O.

After the detection of possible discontinuous entities, the second stage starts assuming all the remaining entities appear as continuous sequences of tokens. To extract continuous entities, an iterative process is carried on over the tag 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 more ambiguous and does not capture much more information since deeper overlappings are not frequent on the training and development collections. Given this, along with the procedure, two in-construction entities are maintained. In each iteration these two entities are created, extended or emitted in accordance to rules defined considering only the previous and the current tag.

2.3. Relation Extraction Model

The most complete information for solving 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 most important information, and discards the misleading [6, 7, 8]. To determine a possible relation between two entities, the system presented uses as input structures derived from the dependency parse tree associated with the target sentence, to obtain information from both the sentence and the entity pair.

According to observations, highlighted entities in the sentence collections are complete nominal phrases (or at least sub constituents of them). Some of the criteria taken into consideration to establish a dependency relation with a header H in a syntactic construction C , is the fact that H could replace C [11]. 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 ¹, we define such subtree corresponding to an entity e , as **relevant tree for e** , and is denoted further on as S_e . The root is called **entity e nucleus**, and is denoted n_e .

Another important definition, vastly used in literature to address this task, is the **dependency path between two tokens** t_1, t_2 . From now on it is going to 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 a score that measures the strength with which the corresponding relation is present between the input entities.

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 on 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.

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

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

$$p = LSTM(P)$$

At the same time, a ChildSum Tree-LSTM [12] is applied independently over S_{e_1} and S_{e_2} (i.e the representations are obtained independently but using the same set of weights).

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

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

¹For simplicity, any notation related to the sentence is gonna be omitted as long as is clear by the context.

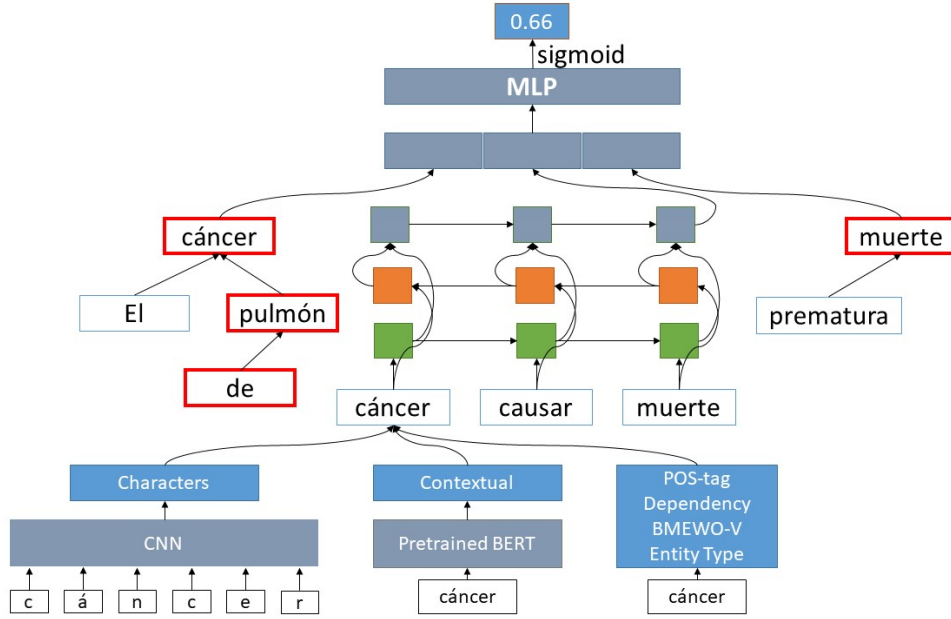


Figure 2: Task B model architecture. Input sentence is *El cáncer de pulmón puede causar muerte prematura* with the highlighted entities *cáncer de pulmón* and *muerte*.

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

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

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 predicted. If none of the scores is greater than such threshold, then no relation is reported. The threshold value is added as an 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.

2.4. Parameters Setup and Training

For both subtasks, the training procedure was carried out using only the training collection provided to contestants.

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:

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

being Z 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:

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

As explained before, the model output does not make use of the fake none relation. A negative sampling strategy is used so that the model is optimized 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 used to compute the character embedding of words. In 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 1 shows the hyperparameter setup for both models.

2.5. Implementation

The systems were implemented using Python programming language, with PyTorch (v1.4.0) library as the deep neural networks framework. BERT-based contextual embeddings were obtained from the `bert-multilingual-uncased` pretrained model, using the Python library `pytorch-pretrained-bert`² (v0.6.2). POS-tag and dependency tree were obtained using the Python library `spacy`³ (v2.2.1), specifically the model `es_core_news_md` (93MB).

Both of the models were trained in a personal computer with the following features: Intel(R) Core(TM) i7-6500 CPU at a frequency of 2.50GHz, with an installed memory of 8.00 GB with no GPU available for CUDA. The total training time for the entity model took less than 5 hours, whereas the relation model was close to 12 hours.

3. Results

The evaluation in both tasks was carried out using the annotated corpus proposed in the challenge. The results were measured with a standard F1 measure as described in detail in the

²<https://pypi.org/project/pytorch-pretrained-bert/>

³<https://spacy.io/>

Table 1
Hyperparameter setup for NER (left) and RE (right) models

Parameter	Value	Parameter	Value
Input embeddings			
Contextual embedding	3072 (last four)	Contextual embedding	768 (last layer)
Character embedding	50	Character embedding	50
POS-tag embedding	50	POS-tag embedding	50
		Dependency embedding	50
		BMEWO-V tags embedding	50
		Entity type embedding	50
Neural Net			
CNN hidden size	100	CNN hidden size	100
Spatial 2D Dropout	0.5	BiLSTM hidden size	100
BiLSTM_1 hidden size	300	Dropout rate	0.2
Dropout_1 rate	0.5	LSTM hidden size	50
BiLSTM_2 hidden size	300	Dropout rate	0.5
Dropout_2 rate	0.5	Tree-LSTM hidden	50
		Dropout rate	0.5
Training			
Optimizer	Adam	Optimizer	Adam
Learning rate	0.001	Learning rate	0.001
Epochs	50	Epochs	30
Total parameters	4681528	Total parameters	689713

challenge overview [1]. Also, precision and recall measures were recorded and presented.

Table 2 presents the official results of the competition, given by the evaluation scenario 1. As shown, our system was ranked as third best, achieving an overall F1 score of 0.625, quite close to the second best, and rather far from the first and fourth ranked systems.

Table 3 shows the results corresponding scenarios 2 and 3, where Task A and B were evaluated independently. With F1 scores of 0.814 and 0.598, our system was able to reach the fourth and second positions on Task A and B evaluation scenarios, respectively.

Finally, an additional transfer-learning scenario was proposed in this edition of the challenge. Scenario 4 evaluates the generalization capabilities of the systems to general-topic domains. Table 4 compares the results of the participant systems in this evaluation scenario.

As can be seen, all systems performed worse in this scenario than in scenario 1, but the ordering remained almost identical. Ours was ranked third as in scenario 1, with a F1 score of 0.547.

Table 2
Scenario 1 results

Team	F1	Precision	Recall
Vicomtech	0.665	0.679	0.652
Talp-UPC	0.626	0.626	0.626
UH-MAJA-KD	0.625	0.634	0.615
IXA-NER-RE	0.557	0.580	0.536
UH-MatCom	0.556	0.716	0.455
SINAI	0.420	0.651	0.310
HAPLAP	0.395	0.458	0.347
baseline	0.395	0.458	0.347
ExSim	0.245	0.312	0.202

Table 3
Scenario 2 (left) and 3 (right) results

Team	F1	Precision	Recall	Team	F1	Precision	Recall
SINAI	0.825	0.844	0.806	IXA-NER-RE	0.633	0.647	0.619
Vicomtech	0.820	0.821	0.820	UH-MAJA-KD	0.598	0.629	0.571
Talp-UPC	0.815	0.807	0.824	Vicomtech	0.583	0.671	0.515
UH-MAJA-KD	0.814	0.820	0.808	Talp-UPC	0.574	0.646	0.517
UH-MatCom	0.794	0.824	0.767	UH-MatCom	0.545	0.682	0.453
IXA-NER-RE	0.691	0.726	0.660	SINAI	0.461	0.627	0.365
HAPLAP	0.541	0.503	0.586	HAPLAP	0.316	0.327	0.305
baseline	0.541	0.503	0.586	ExSim	0.131	0.527	0.075
ExSim	0.314	0.292	0.339	baseline	0.131	0.527	0.075

Table 4
Scenario 4 results

Team	F1	Precision	Recall
Talp-UPC	0.583	0.604	0.563
Vicomtech	0.563	0.594	0.535
UH-MAJA-KD	0.547	0.608	0.498
IXA-NER-RE	0.478	0.563	0.416
UH-MatCom	0.373	0.726	0.250
SINAI	0.281	0.626	0.181
HAPLAP	0.137	0.281	0.091
baseline	0.137	0.281	0.091
ExSim	0.122	0.253	0.080

4. Discussion

The BERT-based contextual embeddings proved to contain useful features for solving both NER and RE tasks. We also experimented using pretrained word embeddings trained on a medical-content corpus extracted from Wikipedia, and the model with the BERT features outperformed

the latter in both tasks. The combined usage of them proved to be rather ineffective. Also, cased and uncased BERT models were tested and the cased model showed better results in the NER model. However, the RE task experienced a rather insignificant decay in performance. This results are based on the model selection process carried out using the development collection.

It is worth mentioning that our experiments showed that BERT is not enough to solve neither of the tasks as described in eHealth-KD 2020 challenge. BERT-only based models (i.e., models with BERT-based inputs and a corresponding CRF or MLP output layers), failed to perform well in both tasks.

For both cases, all the features used as inputs, as described in Section 2, proved to be determinant to achieve top performance. In the particular case of the RE model, aside from BERT-based contextual embeddings, information obtained from the NER task about the entities (i.e., the BMEOW-V tag and the entity type), are the most influential.

Finally, regarding the training process, it is worth noting the fact that the training time of the NER model is significantly shorter than the corresponding to the RE model. This is something expected, since our RE approach defines as a train example a sentence and a pair of entities (thus, much more training examples). Also, as for previous eHealth events, systems still perform significantly poorer in the RE task. These facts might lead to the conclusion that many findings are yet to be done.

5. Conclusions

In this work were described the system proposed by UH-MAJA-KD team at the IberLEF eHealth-KD 2020: eHealth Knowledge Discovery challenge. For Subtask A was proposed a hybrid Stacked-BiLSTM-CRF model, using BERT pretrained contextual embeddings and other syntactic features. This model obtained competitive performance in Scenario 2, where it was located in fourth place with a small difference with respect to the top 3. Task B was addressed in a pairwise-query fashion, encoding information about the sentence and the given pair of entities using syntactic structures derived from the dependency parse tree, and by the means of LSTM-based RNN. The output is obtained scoring every possible relation via a Multilayer Perceptron with a sigmoid activation function. Our model obtained the second place in Scenario 3. The system reached the third position in the overall standing (Scenario 1), and also in the transfer learning scenario (Scenario 4).

It is proposed as future work to fine-tune BERT embeddings along with the training process of the proposed models, looking for contextual embeddings to be trained specifically for these tasks. Also, to use domain specific features like gazetteers looking to improve performance in the health domain. Finally, to develop and evaluate the usage of joint models for solving both tasks.

References

- [1] A. Piad-Morffis, Y. Gutiérrez, H. Cañizares-Díaz, S. Estevez-Velarde, Y. Almeida-Cruz, R. Muñoz, A. Montoyo, Overview of the eHealth Knowledge Discovery Challenge at IberLEF 2020, in: Proceedings of the Iberian Languages Evaluation Forum co-located with

- 36th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2020, Spain, September, 2020., 2020.
- [2] E. Martínez Cámara, Y. Almeida Cruz, M. C. Díaz Galiano, S. Estévez-Velarde, M. Á. García Cumbreiras, M. García Vega, Y. Gutiérrez, A. Montejo Ráez, A. Montoyo, R. Muñoz, et al., Overview of tass 2018: Opinions, health and emotions (2018).
 - [3] A. Piad-Morffis, Y. Gutiérrez, J. P. Consuegra-Ayala, S. Estevez-Velarde, Y. Almeida-Cruz, R. Munoz, A. Montoyo, Overview of the ehealth knowledge discovery challenge at iberlef 2019, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019). CEUR Workshop Proceedings, CEUR-WS. org, 2019.
 - [4] J. Li, A. Sun, J. Han, C. Li, A survey on deep learning for named entity recognition, IEEE Transactions on Knowledge and Data Engineering (2020).
 - [5] J. Lafferty, A. McCallum, F. C. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001).
 - [6] Y. Liu, F. Wei, S. Li, H. Ji, M. Zhou, H. Wang, A dependency-based neural network for relation classification, arXiv preprint arXiv:1507.04646 (2015).
 - [7] S. Zhang, D. Zheng, X. Hu, M. Yang, Bidirectional long short-term memory networks for relation classification, in: Proceedings of the 29th Pacific Asia conference on language, information and computation, 2015, pp. 73–78.
 - [8] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, Z. Jin, 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, 2015, pp. 1785–1794.
 - [9] J. M. Alvarado, E. Q. Caballero, A. Rodriguez, Uh-maja-kd at ehealth-kd challenge 2019 (2019).
 - [10] A. Viterbi, Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, IEEE transactions on Information Theory 13 (1967) 260–269.
 - [11] A. M. Zwicky, Heads, Journal of linguistics 21 (1985) 1–29.
 - [12] K. S. Tai, R. Socher, C. D. Manning, Improved semantic representations from tree-structured long short-term memory networks, arXiv preprint arXiv:1503.00075 (2015).