VSP at eHealth-KD Challenge 2019

Recurrent Neural Networks for Relation Classification in Spanish eHealth Documents

Víctor Suárez-Paniagua

Computer Science Department, Carlos III University of Madrid.
Leganés 28911, Madrid, Spain.
vspaniag@inf.uc3m.es
http://hulat.inf.uc3m.es/en/nosotros/miembros/vsuarez

Abstract. This article describes the proposed system by the VSP team in the Relation Classification subtask of the eHealth-KD Challenge 2019. The architecture is a bidirectional Recurrent Neural Network with LSTM cells (BiLSTM) that uses the word embedding, position embedding and entity type embedding of each word as inputs. Later, a Softmax layer classifies the relationships between entities in the sentences. The presented BiLSTM model reached a 49.33% in F1 for the Scenario 3 of the challenge (Relation Classification). Moreover, the system ranked third out of eight teams that participated in this subtask. The main advantage of this approach is that any hand-crafted feature is used because the system can extract the relevant features automatically.

Keywords: Relation Classification, Deep Learning, Recurrent Neural Network, LSTM, Biomedical Texts

1 Introduction

The number of scientific publications increased until 6% each year, whose largest subject area is the biomedical domain [2]. The manual revision and annotation of all the texts is a very arduous and time-consuming task. However, the extraction of the key information from electronic health (eHealth) documents is vital for health professionals to be up to date. For this reason, the automatic detection and classification of the most relevant words and their relationships can reduce the vast of time for these tasks.

The TASS-2018 Task 3 [5] was the first challenge about the development and evaluation of Natural Language Processing (NLP) systems for extracting the relevant information in Spanish eHealth documents. Following this task, a new edition of this competition was created with an increased number of example in the dataset and the definition of new key phrase and relationship categories [7].

Nowadays, Recurrent Neural Networks (RNN) has shown good performance for tasks where the data are sequential. Concretely, Long-Short Term Memory (LSTM) cells are designed to capture the long distance dependencies between words in the sentences. In Relation Classification, the bidirectional Recurrent Neural Network with LSTM cells (BiLSTM) of [8] improves the results for sentences that involve drug interactions. This model overcomes the previous works based on Convolutional Neural Networks (CNN) in the biomedical domain. The top systems in TASS-2018 Task 3 were based on CNN architectures [6, 9], while RNN architectures were unexplored for the Relation Classification in Spanish eHealth documents.

This work describes the participation of the VSP in the Scenario 3 of the eHealth-KD Challenge 2019 that involves the classification between entities. The proposed architecture is a BiLSTM that generates the representation of the sentences with a relationship and a Softmax layer that classifies the categories of the relationships in one model. The system uses the representation of the words in the sentences together with the information of the entity types labels and the position with respect to the two interacting entities as embeddings.

2 Dataset

An annotated dataset of Spanish eHealth documents was developed for the development and evaluation of the systems in the eHealth-KD Challenge 2019. This corpus was manually extracted from MedlinePlus documents using only the health and medicine topic written in the Spanish language. The data was divided into three datasets: training, development and test sets. The training set and development sets have 600 and 100 manually annotated sentences used for the learning step and the validation of the models, respectively. Additionally, the test set contains 100 sentences together with the annotations of the mentions for the evaluation of the Scenario 3.

The annotated relationship between entities are divide in four categories:

- General relations: indicates general relationships between the entities, they are *is-a*, *same-as*, *has-property*, *part-of*, *causes* and *entails* classes.
- Contextual relations: indicates the refinement of an entity, they are *in-time*, *in-place* and *in-context* classes.
- Action roles: defines the role plays related to an *Action* entity, they are the *subject* and *target* classes.
- Predicate roles: defines the role plays related to an *Predicate* entity, they are the *domain* and *arg* classes.

A more detailed description of the annotation guidelines can be found in [7].

2.1 Pre-processing phase

The relationships in the eHealth-KD Challenge 2019 are asymmetrical, that is the two entities are related in one direction. For this reason, a pair of entities are

annotated with two labels for both directions. Thus, a sentence with n entities will have $(n-1) \times n$ instances. Each instance is labelled with one of the thirteen classes defined by the task. In addition, a *None* class is also considered for the non-relationship between the entities. Table 1 shows the resulting number of instances for each class on the train, validation and test sets.

Table 1. Number of instances for each relationship type in each dataset: train, validation and test.

Label	Train	Validation	Test
is-a	284	50	52
same-as	77	6	11
has-property	81	17	8
part-of	49	1	19
causes	245	24	38
entails	86	14	17
in-time	87	25	12
in-place	258	24	16
in-context	398	64	56
subject	447	62	85
target	955	163	154
domain	179	21	28
arg	197	28	32
None	20341	3077	3012

Once the instances are extracted from the documents, the sentences are tokenized and cleaned similarly to [3], converting the numbers to a common name, words to lower-case, replacing special Spanish accents to Unicode, e.g $\tilde{n}$ to n, and separating special characters with white spaces by regular expressions. Besides, the two target entities of each instance are replaced by the labels "*entity1*", "*entity2*", and by "*entity0*" for the remaining entities. This method blinds the mentions in the instance for the generalization of the model.

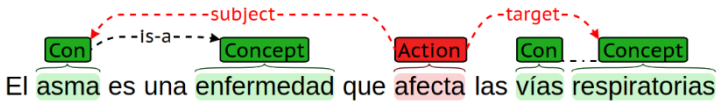

Fig. 1. The entities and their relationships in the sentence *'El asma es una enfermedad que afecta las vías respiratorias.'*. English translation: *'Asthma is a disease that affects the respiratory tract.'*.

Figure 1 shows the sentences *'El asma es una enfermedad que afecta las vías respiratorias.'* where the entities *asma*, *enfermedad*, *afecta* and *vías respiratorias* should be transformed to each relation instances (see Table) 2.

Table 2. Instances with two different entities relationship after the pre-processing phase with entity blinding of the sentence *'El asma es una enfermedad que afecta las vías respiratorias.'*.

Relationship between entities	Instances after entity blinding	Label
(asma → enfermedad)	*'el entity1 es una entity2 que entity0 las entity0 .'*	*is-a*
(asma ← enfermedad)	*'el entity2 es una entity1 que entity0 las entity0 .'*	*None*
(asma → afecta)	*'el entity1 es una entity0 que entity2 las entity0 .'*	*None*
(asma ← afecta)	*'el entity2 es una entity0 que entity1 las entity0 .'*	*subject*
(asma → vías respiratorias)	*'el entity1 es una entity0 que entity0 las entity2 .'*	*None*
(asma ← vías respiratorias)	*'el entity2 es una entity0 que entity0 las entity1 .'*	*None*
(enfermedad → afecta)	*'el entity0 es una entity1 que entity2 las entity0 .'*	*None*
(enfermedad ← afecta)	*'el entity0 es una entity2 que entity1 las entity0 .'*	*None*
(enfermedad → vías respiratorias)	*'el entity0 es una entity1 que entity0 las entity2 .'*	*None*
(enfermedad ← vías respiratorias)	*'el entity0 es una entity2 que entity0 las entity1 .'*	*None*
(afecta → vías respiratorias)	*'el entity0 es una entity0 que entity1 las entity2 .'*	*target*
(afecta ← vías respiratorias)	*'el entity0 es una entity0 que entity2 las entity1 .'*	*None*

In the corpus, there are some instances with multiple annotations, the vast of them are annotated like *target* and *subject*. Only one of these classes is kept because the proposed system does not tackle the multi-class classification. Additionally, there are entities with discontinuous tokens that have some overlapping parts with other entities. Thus, only the non-overlapping parts of the mentions are kept given that the entity blinding process cannot deal with this kind of entities.

3 BiLSTM model

This section presents the BiLSTM architecture which is used for the Scenario 3 of the eHealth-KD Challenge 2019. Figure 2 shows the RNN model where the inputs are the preprocessed sentences and it generates a prediction label of the relationship between the marked entities.

3.1 Word table layer

Firstly, the preprocessed sentences are transformed into an input matrix for the RNN architecture. Padding is added to all the sentences until reaching the maximum length of a sentence in the dataset (denoted by n). Thus, the sentences shorter than n are padded with an auxiliary token "*0*".

Each word in the sentences is represented by its word and entity type embeddings. These embeddings are extracted from the word embedding matrix $\mathbf{W}_e \in \mathbb{R}^{|V| \times m_e}$ where V is the vocabulary size and m_e is the word embedding

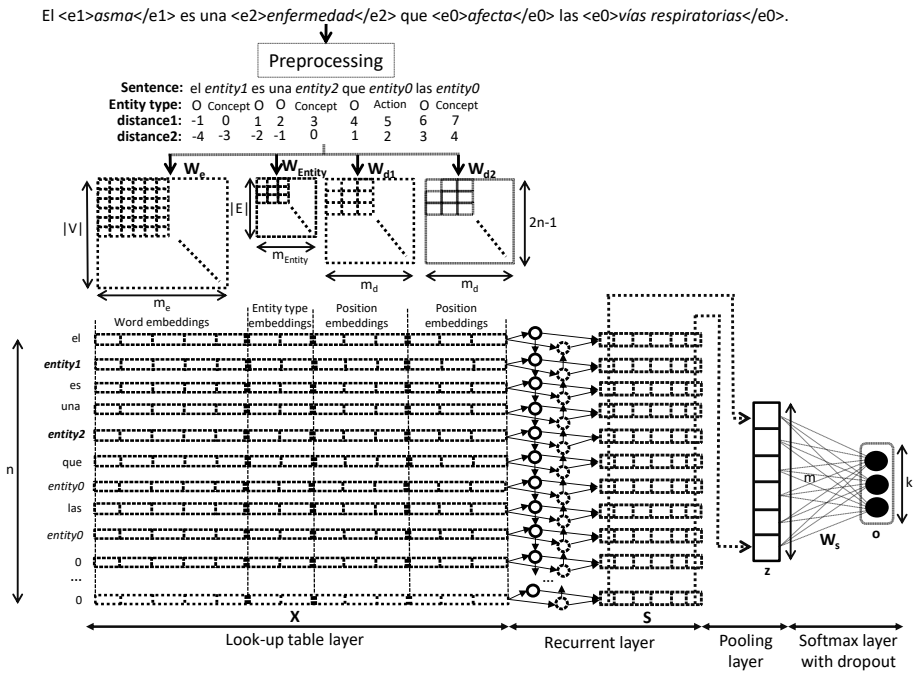

Fig. 2. BiLSTM model for the Relation Classification task in the eHealth-KD Challenge 2019.

dimension and the entity embedding matrix $\mathbf{W}_{Entity} \in \mathbb{R}^{|E| \times m_{Entity}}$ where E is the vocabulary size of the entity types and m_{Entity} is the entity type embedding dimension, respectively.

In addition, two position embeddings are concatenated to these vector in order to represent the relative position of each word with respect to the two interacting mentions. These distances are mapped into a real value vectors with the two position embedding matrices, $\mathbf{W}_{d1} \in \mathbb{R}^{(2n-1) \times m_d}$ and $\mathbf{W}_{d2} \in \mathbb{R}^{(2n-1) \times m_d}$ where m_d is the position embedding dimension.

Finally, each word in the sentence is described by a vector giving a matrix that represents the sentence of the relationship $\mathbf{X} \in \mathbb{R}^{n \times (m_e + m_{Entity} + 2m_d)}$.

3.2 Recurrent layer

The resulting matrix is the input for the Recurrent layer. In this system, LSTM cells [1] are implemented in the RNN. This kind of cells defines a gating mechanism for creating a word representation taking the information of the current and previous cells. The input gate i_t, the forget gate f_t and the output gate o_t for the current t step transform the input vector x_t taking the previous output h_{t-1} using its corresponding weights and bias computed with a sigmoid function. The cell state c_t takes the information given from the previous cell state c_{t-1}

regulated by the forget cell and the information given from the current cell c'_t regulated by the input cell using the element-wise represented as:

$$
\begin{aligned}
f_t &= \sigma(\mathbf{W}_f \cdot [h_{t-1}, x_t] + b_f) \\
i_t &= \sigma(\mathbf{W}_i \cdot [h_{t-1}, x_t] + b_i) \\
c'_t &= tanh(\mathbf{W}_c \cdot [h_{t-1}, x_t] + b_c) \\
c_t &= f_t * c_{t-1} + i_t * c'_t \\
o_t &= \sigma(\mathbf{W}_o \cdot [h_{t-1}, x_t] + b_o) \\
h_t &= o_t * tanh(c_t)
\end{aligned}
$$

where $\mathbf{W}_f$, $\mathbf{W}_i$, $\mathbf{W}_c$ and $\mathbf{W}_o$ are the weights and b_f, b_i, b_c and b_o are the bias term of each gate.

Later, the current output h_t is represented with the hyperbolic function of the cell state and controlled by the output gate. Furthermore, another Recurrent layer can be applied in the other direction from the end of the sequence to the starting word. Computing the two representations is beneficial for extracting the relevant features of each word because they have dependencies in both directions. Finally, the output vector of both directions is concatenated giving an output matrix $\mathbf{S} \in \mathbb{R}^{n \times m}$ where m is the number of output dimensions for the Recurrent layer.

3.3 Pooling layer

The pooling layer extracts the most relevant features of the output matrix in the Recurrent layer using an aggregating function. In this model, the max function is selected to produce a single value for each output as $z_f = max\{\mathbf{s}\} = max\{s_1; s_2; ...; s_n\}$. Thus, the vector $\mathbf{z} = [z_1, z_2, ..., z_m]$ is created, whose dimension is the total number of filters m representing the relation instance.

3.4 Softmax layer

Before the classification, a dropout is applied to prevent overfitting. Thus, a reduced vector $\mathbf{z}_d$ is obtained from randomly setting the elements of $\mathbf{z}$ to zero with a probability p following a Bernoulli distribution. After that, this vector is fed into a fully connected Softmax layer with weights $\mathbf{W}_s \in \mathbb{R}^{m \times k}$ to compute the output prediction values for the classification as $\mathbf{o} = \mathbf{z}_d \mathbf{W}_s + d$ where d is a bias term; in this case, there are $k = 13$ classes in the dataset and the "*None*" class. At test time, the vector $\mathbf{z}$ of a new instance is directly classified by the Softmax layer without a dropout.

3.5 Learning

The following BiLSTM parameters $\theta = (\mathbf{W}_e, \mathbf{W}_{Entity}, \mathbf{W}_{d1}, \mathbf{W}_{d2}, \mathbf{W}_s, d, \overrightarrow{\mathbf{W}}_f, \overrightarrow{b}_f, \overrightarrow{\mathbf{W}}_i, \overrightarrow{b}_i, \overrightarrow{\mathbf{W}}_c, \overrightarrow{b}_c, \overrightarrow{\mathbf{W}}_o, \overrightarrow{b}_o, \overleftarrow{\mathbf{W}}_f, \overleftarrow{b}_f, \overleftarrow{\mathbf{W}}_i, \overleftarrow{b}_i, \overleftarrow{\mathbf{W}}_c, \overleftarrow{b}_c, \overleftarrow{\mathbf{W}}_o, \overleftarrow{b}_o)$ need to

be learned in training the network. To this end, the conditional probability of a relation r obtained by the Softmax operation as

$$p(r|\mathbf{x},\theta) = \frac{\exp(\mathbf{o}_r)}{\sum_{l=1}^{k}\exp(\mathbf{o}_l)}$$

is minimized by the negative log-likelihood function for all instances ($\mathbf{x}_i$,y_i) in the training set T as follows

$$J(\theta) = -\sum_{i=1}^{T}\log p(y_i|\mathbf{x}_i,\theta)$$

In addition, the stochastic gradient descent over shuffled mini-batches and the Adam update rule [4] minimizes the objective function to learn the parameters.

4 Results and Discussion

The weights of the BiLSTM model were learned on the training set during 25 epochs using mini-batches and Adam update rule [4], while the best model was chosen with the best performance on the validation set. Table 3 shows the model parameters and their values fine-tuned for the classification of relationships between entities in Spanish eHealth documents. The embeddings of the words, the entity types and the two positions were randomly initialized and learned during the training of the network.

Table 3. The BiLSTM model parameters and their values used for the results.

Parameter	**Value**
Maximal length in the dataset, n	66
Word embeddings dimension, M_e	300
Position embeddings dimension, M_d	10
Entity type embeddings dimension, M_d	10
LSTM output dimension, m	200
Dropout rate, p	0.5
Mini-batch size	50
Optimizer	Adam
Learning rate	0.001

The results were measured with precision (P), recall (R) and F1, defined as:

$$P = \frac{C}{C+S} \quad R = \frac{C}{C+M} \quad F1 = 2\frac{P\times R}{P+R}$$

where Correct (C) are the relations that matched to the test set and the prediction, Missing (M) are the relations that are in the test set but not in the prediction, and Spurious (S) are the relations that are in the prediction but not in the test set.

Table 4 presents the results of the BiLSTM for all the classes and the final results in the Scenario 3 (Relation Classification). It can be observe that the number of Missing is higher compared to the number of Spurious which makes a low Recall in almost all the classes. In general the Action and Predicate roles are classified better than the General or the Contextual relations. In addition, the classes with less than 100 examples in the training set, such as *same-as*, *has-property*, *part-of*, *entails* or *in-time*, has a low performance because the architecture can not extract the relevant features of these classes compared with the more than 20,000 examples of the *None* class. Thus, the vast of them are classified as *None* and taken as Missing.

Table 4. Results over the test set using a BiLSTM model.

Label	**C**	**M**	**S**	**P**	**R**	**F1**
is-a	24	32	13	64.86%	42.86%	51.61%
same-as	2	11	1	66.67%	15.38%	25%
has-property	2	7	5	28.57%	22.22%	25%
part-of	3	16	2	60%	15.79%	25%
causes	5	43	5	50%	10.42%	17.24%
entails	4	14	4	50%	22.22%	30.77%
in-time	3	8	9	25%	27.27%	26.09%
in-place	4	12	23	14.81%	25%	18.6%
in-context	24	33	11	68.57%	42.11%	52.17%
subject	38	59	18	67.86%	39.18%	49.67%
target	98	61	59	62.42%	61.64%	62.03%
domain	15	18	11	57.69%	45.45%	50.85%
arg	19	13	7	73.08%	59.38%	65.52%
Scenario 3	241	327	168	58.92%	42.43%	49.33%

Eight teams participated in this subtask being the BiLSTM model the third highest F1. The performance of the proposed system is very promising obtaining 49.33% in F1 as official results in Scenario 3 of the eHealth-KD Challenge 2019 that has thirteen relation categories. One of the main advantage of this approach is that it does not require any external knowledge resource.

5 Conclusions and Future work

This paper presents a BiLSTM model for the eHealth-KD Challenge 2019 Scenario 3 (Relation Classification of Spanish eHealth documentss). The official results for the proposed system are very promising because the model is a simple architecture that does not need expert domain knowledge or external features. However, the performance of the method is very low in Recall measure because there are a high number of Missing instances that are classified as the *None* class. One possible solution could be the creation of four independent classification systems for the four kind of classes taking into consideration the entity types of the mentions. Thus, each system could be more balanced and specialized in each type of relationships. Moreover, it is hard for the system detecting the directionality of the relationships. For this reason, the creation of a reverse class for each relation type could help to the architecture in order to distinguish the direction of the relation.

As future work, the aggregation of external feature as embedding vector, such as Part-of-Speech tags, semantic tags or syntactic parse trees, could improves the representation of each word and increase the performance. Furthermore, the exploration of deeper layers in the Recurrent layer is proposed to be included in the BiLSTM model for the classification of relationships in Spanish eHealth documents.

References

1. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
2. Johnson, R., Watkinson, A., Mabe, M.: The stm report, an overview of scientific and scholarly publishing. (2018)
3. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1746–1751 (2014)
4. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014)
5. Martínez-Cámara, E., Almeida-Cruz, Y., Díaz-Galiano, M.C., Estévez-Velarde, S., García-Cumbreras, M.A., García-Vega, M., Gutiérrez, Y., Montejo-Ráez, A., Montoyo, A., Muñoz, R., Piad-Morffis, A., Villena-Román, J.: Overview of TASS 2018: Opinions, health and emotions. In: Martínez-Cámara, E., Almeida Cruz, Y., Díaz-Galiano, M.C., Estévez Velarde, S., García-Cumbreras, M.A., García-Vega, M., Gutiérrez Vázquez, Y., Montejo Ráez, A., Montoyo Guijarro, A., Muñoz Guillena, R., Piad Morffis, A., Villena-Román, J. (eds.) Proceedings of TASS 2018: Workshop on Semantic Analysis at SEPLN (TASS 2018). CEUR Workshop Proceedings, vol. 2172. CEUR-WS, Sevilla, Spain (September 2018)
6. Medina, S., Turmo, J.: Joint classification of key-phrases and relations in electronic health documents. Tech. rep., Sevilla, Spain (September 2018), `http://ceur-ws.org/Vol-2172/p9-talp_tass2018.pdf`
7. Piad-Morffis, A., Gutiérrez, Y., Consuegra-Ayala, J.P., Estevez-Velarde, S., Almeida-Cruz, Y., Muñoz, R., Montoyo, A.: Overview of the eHealth Knowledge Discovery Challenge at IberLEF 2019 (2019)

8. Sahu, S.K., Anand, A.: Drug-drug interaction extraction from biomedical texts using long short-term memory network. Journal of Biomedical Informatics **86**, 15 – 24 (2018). https://doi.org/https://doi.org/10.1016/j.jbi.2018.08.005, http://www.sciencedirect.com/science/article/pii/S1532046418301606
9. Suarez Paniagua, V., Segura Bedmar, I., Martínez Fernández, P.: Labda at tass-2018 task 3: Convolutional neural networks for relation classification in spanish ehealth documents. Tech. rep., Sevilla, Spain (September 2018), http://ceur-ws.org/Vol-2172/p7-labda_tass2018.pdf