UH-MAJA-KD at eHealth-KD Challenge 2019

Deep Learning Models for Knowledge Discovery in Spanish eHealth Documents

Jorge Mederos Alvarado[1], Ernesto Quevedo Caballero[1], Alejandro Rodríguez Pérez[1], and Rocío Cruz Linares[1][0000−0002−0069−8950]

University of Havana, Cuba
http://www.uh.cu

Abstract. This paper describes the solution presented by the *UH-MAJA-KD* team in IberLEF eHealth-KD 2019: eHealth Knowledge Discovery challenge. Separate strategies were developed to solve substasks A and B, both based on deep learning models using domain-specific word embeddings, and architectures using Bidirectional Long-Short Term Memory (BiLSTM) cells. In the case of Subtask A, Conditional Random Field was used to produce an output in BMEWO-V tag system to extract keyphrases. For Subtask B, two stacked BiLSTM layers are used along with Shortest Dependency Path in-between a pair of keyphrases to determine possible relationships between them.

Keywords: eHealth · Knowledge discovery · Keyphrase extraction · Keyphrase classification · Relationships extraction

1 Introduction

In the health domain, the large number of research and publications every year makes nearly impossible for doctors and biomedical researchers to keep up to date with the literature in their fields. Thus, finding ways to effectively manage the vast amounts of information and extract knowledge from it is really important nowadays. This could help in the task of obtaining new and better scientific results or in the diagnosis of complex diseases. Due to all of these reasons, a high interest around the scientific community has aroused in developing systems to automatically extract knowledge from medical texts.

There is an increasing amount of efforts oriented towards this direction. One of them is the IberLEF eHealth-KD 2019: eHealth Knowledge Discovery challenge [8], in which context this paper was developed. The goal of this challenge was the discovery of knowledge in medical texts, via the extraction and classification of keyphrases, as well as the determination of semantic relationships between pairs of keyphrases. The challenge was divided into two subtasks: A and

B, one for keyphrase extraction and classification, and the other oriented to the extraction of semantic relationships.

This paper describes the solution presented by the *UH-MAJA-KD* team in IberLEF eHealth-KD 2019: eHealth Knowledge Discovery challenge. It proposes a strategy using a hybrid model that combines a Bidirectional Long Short Memory (BiLSTM) layer with a Conditional Random Field (CRF) layer for Subtask A. This model is inspired on the model presented by UCM team [10] in the past edition of the challenge; in addition, domain-specific word embeddings are used. For Subtask B a multiclass classifier is proposed, taking as input a sequence of features vectors of the tokens in the Shortest Dependency Path between pairs of keyphrases.

The rest of the paper is organized as follows. In section 2 is given a brief overview of word embeddings, and the particular one used along the rest of the paper. Sections 3 and 4 describe specifically the approach to solve Subtasks A and B respectively. Then, the results of the models proposed are presented in section 5, and finally, brief conclusions and future work lines are presented in section 6.

2 Word embeddings

Word embeddings are a strategy to represent words as real numbers vectors on a reduced-dimension space. It is desired for these vectors to have the property of context similarity, this is, for words that appear commonly in the same context, their respective vectors must be close in the embedding space, under some distance measure. There are many methods to obtain such embeddings in literature, most of them based on probabilistic models and/or neural networks. Among most popular are found *word2vec* [5], *fastText* morphological representation [1] and *GloVe* (Global Vectors for Word Representations) [7].

Regarding neural network-based word embeddings, the corpus used to train them is crucial in its performance, precisely because the corpus determines the words and contexts in which the words appear. Intuitively, domain-specific corpora should be better at showing contextual and semantic relations regarding that specific domain. Consequently, a corpus was built based on Spanish Wikipedia [1], extracting medical content pages. The corpus size is of approximately 27 million words, with essentially medical content. To capture domain-specific semantic and contextual information, a word embedding was trained on this corpus. To do this, it was used the *word2vec* algorithm API offered by **gensim** [9] python library, using the architecture CBOW (Continuous Bag of Words) [2]. Embedding details are shown next:

- Embedding space dimensions: 300.
- Windows size: 5.
- Vocabulary size: approximately 500 thousand words.
- Negative sample: 5

[1] es.wikipedia.org

3 Subtask A

The goal of Subtask A was to extract keyphrases from sentences and to classify them as Concept, Action, Reference or Predicate. The proposed solution splits this subtask into four more specific ones, each of those to extract and classify concepts, actions, references and predicates respectively. The defined architecture is the same in all the four cases, but each model is trained independently, using as training examples only those of its corresponding task (e.g the model that extracts and classifies keyphrases in Concept, only receives as input annotations of Concept keyphrases). This is done in order to improve specific weight learning for each type of keyphrase since they could be under different hypothesis functions, making difficult to the model learning 'good' weights for all of them together. Moreover, to process them united could lead to more ambiguity in the decoding process (which will be explained at 3.3), making more solutions unfeasible. Finally, all the keyphrases detected by all the four models are put together.

3.1 Model Input

The system receives as input a sentence string, thus it needs some preprocessing to build an appropriated input to the models. The first step is to tokenize the sentences as all model inputs expect a sequence of tokens.

For each token in which the sentence was split, the input for that token consist of a list of three feature vectors:

. **Character encodings:** Concatenation of one-hot encoded vectors of the characters contained in the word.
. **PoS-tag vector:** One hot encoded vector of Part of Speech (PoS) information.
. **Word indexes:** One hot encoded index in the word embedding vocabulary.

To obtain the first standard ASCII alphabet was used. To extract PoS-tag information the python library **spacy**[2] was used. In the case of the third input, some words are captured using regular expressions and substituted with special tokens defined in the word embedding vocabulary (e.g currencies, units of measurement and other words with digits or non-latin characters). In the case of words not appearing in the vocabulary, a special token *'unseen'* was defined.

3.2 Model Architecture

Each of the four models used to solve the Subtask A receives a sequence of token inputs as described in 3.1, and produces a same sized sequence with labels for each token in the BMEWO-V tagging system which will be described in the section below.

The architecture is conformed by four main components:

[2] spacy.io

. Word embedding matrix
. Char embedding BiLSTM [3]
. Token-level BiLSTM
. CRF classifier [4]

It is pipelined as follows. For each token in the input sequence, the pre-trained word embedding layer produces an embedding vector using the word index input. The character embedding layer receives the sequence of character encodings contained in the word and produces a vector, capturing character level information for each word. These two vectors are concatenated with the PoS-tag vector information of the word, and all together serve as input to each time step of the token-level BiLSTM layer. Finally, the outputs of the BiLSTM layer are passed to a CRF layer.

A summary of the model is shown in Figure 1.

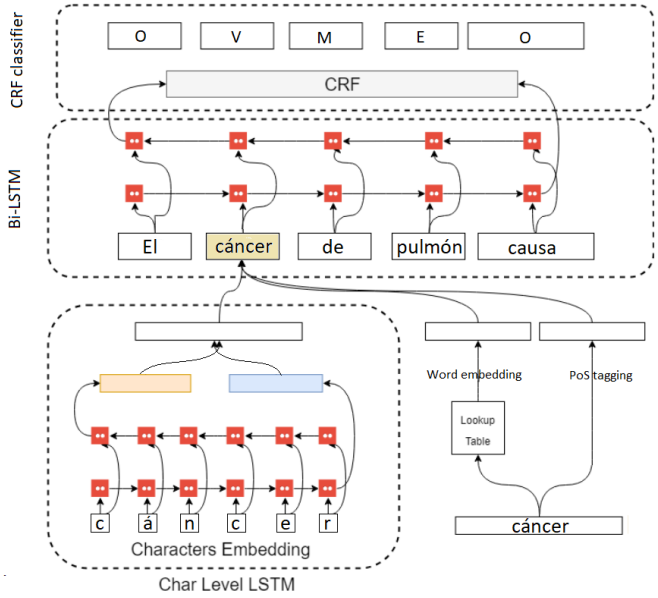

Fig. 1. Subtask A model summary

3.3 Postprocessing

The CRF layer produces a sequence of tags in the BMEWO-V tagging system. This classification corresponds to **B** for begin of a keyphrase, **M** for medium, **E** for end, **W** for tokens that are a keyphrase themselves and **O** for tokens that do

not represent anything. It also takes into account the possibility of keyphrases overlapping, including the tag **V** in such cases. For the sentence: *El cáncer de pulmón causa muerte prematura*, the model detecting Concept keyphrases should produce the output: **O-V-M-E-O-B-E**.

Since the expected output in Subtask A is a sequence of keyphrases for each sentence, a procedure is necessary to transform the BMEWO-V tag sequence got from a given sentence, in a keyphrase sequence corresponding to the output expected in Subtask A. This process was called decoding. There is an important challenge in this process: tokens belonging to a keyphrase are not necessarily continuous in the sentence. Taking this into account, the decoding process is divided into two stages. First, discontinuous keyphrases are detected and then, at a second moment, continuous keyphrases.

In accordance to Spanish correct use, The set of tag sequences that must be interpreted as a group of discontinuous keyphrases were reduced to those that match the regular expressions *(V+)((M*EO*)+)(M*E)* and *((BO)+)(B)(V+)*. The first one corresponds to keyphrases that share their initial tokens, and the second one to those that share their final tokens. These two capture most of the desired discontinuous keyphrases. Among the examples of the first case it is found the fragment *cáncer de pulmón y de mama*, tagged as V-M-E-O-M-E, where keyphrases *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 keyphrases *tejidos humanos* and *órganos humanos* are found. When a match is detected and the keyphrases are extracted, all the tags in that fragment are set to tag O.

After the detection of possible discontinuous keyphrases, the second stage starts assuming all the remaining keyphrases appear as continuous sequences of tokens. To extract continuous keyphrases, 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 is not common in Spanish to find examples with deeper overlapping. Given this, along with the procedure, two in-construction keyphrases are maintained. In each iteration these two keyphrases are *created*, *extended* or *emitted* in accordance to rules defined considering only the previous and the current tag. Tag **B** indicates to start a new keyphrase, **M** the extension of an existent keyphrase and **E** its ending. Tag **V** introduces overlapping, hence this is the one that causes that there could be two in-construction keyphrases at a given moment. Tag **W** causes the current token to be reported automatically as a keyphrase.

4 Subtask B

The goal of Subtask B was to detect semantic relationships between pairs of keyphrases. The solution proposed consists of traversing every pair of keyphrases and determine whether one of the defined semantic relationships is established

between them or not, via a multiclass classifier. This is accomplished by building a dependency tree for the tokens in the sentence and finding the shortest path in-between the keyphrases along this tree. This is called Shortest Dependency Path [6]. The model is agnostic to any restrictions defined on the relations domain (e.g it is not told in advance that for relation Subject, one of the keyphrases should be an Action), needing to learn it by itself.

4.1 Model Input

Similar to Subtask A models, this model expects a sequence of tokens. For each token in that sequence, the input for that token consists of a list of four feature vectors:

- **Word indexes:** One hot encoded index in the word embedding vocabulary.
- **Syntactic dependency relation vector:** One hot encoded vector of syntactic dependency information.
- **BMEWO-V tag encoding:** One hot encoded BMEWO-V tag.
- **Subtask A type of keyphrase encoding:** One hot classification on Concept, Action, Reference or Predicate of the keyphrase to which token belongs.

The word indexes are obtained as described in 3.1. To extract syntactic dependency information the python library **spacy** was used. The third and fourth inputs are obtained from Subtask A if they were pipelined as in the case of Scenario 1 in the challenge.

4.2 Model Architecture

The architecture is conformed by three main components:

- Word embedding matrix
- Stacked BiLSTMs
- Two dense multiclass classifiers

It is pipelined as follows. For each token in the input sequence, the pre-trained word embedding layer produces an embedding vector using the word index input. The embedding vector is then concatenated with the other three input vectors, and all together serve as input to each time step of the stacked BiLSTM layers. Finally, the last time step output of the stacked BiLSTM layers serves as input of two Dense layers serving as multiclass classifiers, one for each direction in which relationships could be established between the pair of keyphrases, since those are not symmetric.

A summary of the model is shown in Figure 2.

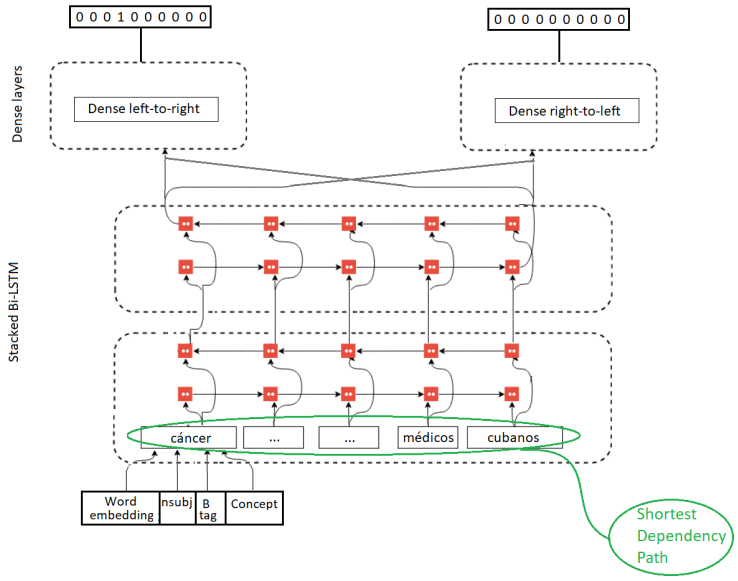

Fig. 2. Subtask B model summary

5 Results

The evaluation in both subtasks was carried out using the annotated corpus proposed in the challenge. The results were measured with precision, recall and F1 in three scenarios as described in the details of IberLEF eHealth-KD 2019: eHealth Knowledge Discovery [8].

Tables 1, 2 and 3 show the results obtained by participants in Scenarios 1,2 and 3 respectively. Scenario 2 measures the results in Subtask A and Scenario 3 only in Subtask B, whereas Scenario 1 combines both Subtask A and B.

As can be observed, the proposal for Subtask A had a competitive performance, being only 0.0047 points lower than the first place in F1 score. However, results on Subtask B are not as promising. The first place critically outperformed the model proposed for Subtask B.

In the case of Subtask A, the model showed faster convergence when training on both Action and Reference labels. This is probably because of the syntactic patterns they show, that are rapidly captured by the model.

It is worth to mention the evaluations that were made on the BMEWO-V decoder. It turned to be over 99% in both precision and recovery when evaluated on perfectly annotated labels. It showed, however, a non-linear decline in performance when evaluated on inaccurately-classified labels.

The set of parameters and the hyper-parameters used to test the models are the following:

Table 1. Scenario 1 results

Scenario 1	F1	Precision	Recall
TALP	0.6394	0.6506	0.6286
coin_flipper(ncatala)	0.6218	0.7454	0.5334
LASTUS-TALN (abravo)	0.5816	0.7740	0.4658
NLP_UNED	0.5473	0.6561	0.4695
Hulat-TaskAB	0.5413	0.7734	0.4163
UH-MAJA-KD	**0.5189**	**0.5644**	**0.4802**
lsi2_uned	0.4934	0.7397	0.3702
IxaMed(iakesg)	0.4869	0.6896	0.3763
_baseline	0.4309	0.5204	0.3677
Hulat-TaskA(jlcuad)	0.4309	0.5204	0.3677
VSP	0.4289	0.4551	0.4056

Table 2. Scenario 2 results

Scenario 2	F1	Precision	Recall
TALP	0.8203	0.8073	0.8336
LASTUS-TALN (abravo)	0.8167	0.7997	0.8344
UH-MAJA-KD	**0.8156**	**0.7999**	**0.8320**
Hulat-TaskA(jlcuad)	0.7903	0.7706	0.8111
coin_flipper (ncatala)	0.7873	0.7986	0.7763
Hulat-TaskAB	0.7758	0.7500	0.8034
NLP_UNED(lsi_uned)	0.7543	0.8069	0.7082
lsi2_uned	0.7315	0.7817	0.6873
IxaMed(iakesg)	0.6825	0.6567	0.7105
_baseline	0.5466	0.5129	0.5851
VSP	0.5466	0.5129	0.5851

Subtask A models:

- Words embeddings dimension: 300
- Characters embeddings dimension: 25
- BiLSTM dimension(Both char level and token level): 64
- BiLSTM dropout and recurrent dropout(Both char level and token level): 0.2
- Optimizer: adam
- Epochs: 30(Concept), 10(Action), 20(Predicate), 10(Reference)

Subtask B model:

- Words embeddings dimension: 300
- First BiLSTM dimension: 64
- Recurrent dropout: 0.4
- Second BiLSTM dimension: 32
- Recurrent dropout: 0.2
- Optimizer: SGD(Nesterov, momentum = 0.9)
- Epochs: 20

Table 3. Scenario 3 results

Scenario 3	**F1**	**Precision**	**Recall**
TALP	0.6269	0.6667	0.5915
NLP_UNED(lsi_uned)	0.5337	0.6235	0.4665
VSP	0.4933	0.5892	0.4243
coin_flipper (ncatala)	0.4931	0.7133	0.3768
IxaMed(iakesg)	0.4356	0.5195	0.3750
UH-MAJA-KD	**0.4336**	**0.4306**	**0.4366**
LASTUS-TALN (abravo)	0.2298	0.1705	0.3521
_baseline	0.1231	0.4878	0.0704
Hulat-TaskAB	0.1231	0.4878	0.0704
Hulat-TaskA(jlcuad)	0.1231	0.4878	0.0704
lsi2_uned	0.1231	0.4878	0.0704

The number of epochs was selected empirically, based on the fast convergence of the models, tending to quickly overfit on training dataset, even though validation data was used. The remaining parameters were selected as standard for similar applications in literature.

6 Conclusions and Future Work

In this work were described the models presented by the *UH-MAJA-KD* team for the IberLEF eHealth-KD 2019: eHealth Knowledge Discovery.

In Subtask A a hybrid BiLSTM and CRF model with specific domain pre-trained word embeddings was proposed. Our model obtained the third place in the Scenario 2. In Subtask B a multiclass classifier using Shortest Dependency Path with pre-trained word embeddings in a specific domain was proposed. Our model obtained the sixth place in the Scenario 3. Our team reached the sixth position in the overall competition standing.

The corpus in which the domain-specific word embedding was trained is relatively small. It is proposed as future work to build a more expressive and abundant corpus to improve the word embedding performance. Also, could be promising to try to concatenate both domain-specific and general purpose word embeddings, in order to gain one's specificity and the generalization capability of the latter. To improve the capabilities of the system in the overall task, it could be convenient to train the system (l.e both models) as a whole, providing Subtask B with the output from Subtask A, needing the first to deal with the errors produced by the latter.

Acknowledgments

We would like to acknowledge the joint project *Tec-UH* of Tecnomática[3] enterprise and the Artificial Intelligence Group at the University of Havana, to allow

[3] https://www.cupet.cu/footer/informatica-automatica-y-comunicaciones/

us to use high-performance computational equipment to develop and test our ideas.

References

1. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017)
2. Gulli, A., Pal, S.: Deep Learning with Keras. Packt Publishing Ltd (2017)
3. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
4. Lafferty, J., McCallum, A., C.N Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001)
5. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International conference on machine learning. pp. 1188–1196 (2014)
6. Li, F., Zhang, M., Fu, G., Ji, D.: A neural joint model for entity and relation extraction from biomedical text. BMC Bioinformatics **18** (12 2017). https://doi.org/10.1186/s12859-017-1609-9
7. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
8. 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)
9. Rehurek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. Citeseer (2010)
10. Zavala, R.M.R., Martınez, P., Segura-Bedmar, I.: A hybrid bi-lstm-crf model for knowledge recognition from ehealth documents. In: Proceedings of TASS 2018: Workshop on Semantic Analysis at SEPLN (TASS 2018) (2018)