E2EJ: Anonymization of Spanish Medical Records using End-to-End Joint Neural Networks

Mohammed Jabreel[1], Fadi Hassan[2], Najlaa Maarrof[1], David Sánchez[2], Josep Domingo-Ferrer[2], and Antonio Moreno[1]

[1] iTAKA: Intelligent Technologies for Advanced Knowledge Acquisition. Department of Computer Science and Mathematics
[2]CYBERCAT-Center for Cybersecurity Research of Catalonia. UNESCO Chair in Data Privacy.
Universitat Rovira i Virgili, Av. Països Catalans 26, E-43007 Tarragona, Catalonia
{mohammed.jabreel, fadi.hassan, najlaa.maaroof, david.sanchez, josep.domingo, antonio.moreno}@urv.cat

Abstract. This paper describes E2EJ, the system that we have developed to participate in the Medical Document Anonymization challenge in the shared task of IberLEf2019. E2EJ is a data-driven and end-to-end neural network. It does not rely on external resources such as part-of-speech tagger. It proposes to solve two problems jointly; the first problem is to automatically identify whether a token is sensitive, whereas the second one is to identify the type of the token. E2EJ shows comparable results to the state-of-the-art systems and outperform the baseline systems. The F1 score of our system on the test set is 96.61% and 95.83% for the sensitivity detection and the token type identification tasks respectively.

Keywords: Anonymization · CRF · Medical Documents. · Deep Learning

1 Introduction

Patient notes in electronic health records (EHRs) contain critical information that may be useful for medical investigations. However, due to privacy concerns, the vast majority of medical investigators can only access anonymized or de-identified notes to protect the confidentiality of patients [1]. Anonymization can be either manual or automated. Manual anonymization means that human annotators label protected health information (PHI). This approach has some drawbacks. First, only a limited set of individuals is allowed to access the identified patient notes. Thus, the task cannot be crowd-sourced. Second, humans are prone to mistakes. Third, manual anonymization is impractical given the size of EHR databases. Therefore, a reliable automated anonymization system would consequently be of high-value [14, 8]. In the literature, there are many

systems for EHR anonymization, which we can categorize them as rule-based, feature-engineering-based, or deep-learning-based approaches.

Starting by a seed collection of sensitive tokens, the idea of rule-based systems is to manually engineer some rules based on regular expressions, syntactic, or dependency structures to expand the collection iteratively [13, 9].

The feature-engineering-based systems aim to train a sequence tagger with rich, hand-crafted features based on linguistic or syntactic information from annotated corpus to predict a label (e.g., O, $B- < entity >$ or $I- < entity >$) on each token in a sentence [5].

Rule-based and feature-engineering-based approaches are labor-intensive for constructing rules or features using linguistic and syntactic information. Despite some promising results, there are two main issues with these approaches. First, the engineering of rules and features is a time-consuming task. Moreover, rules always need to be updated. Second, the systems of these two categories are dependent on some external requirements like a parser analyzing the syntactic and dependency structure of sentences. Therefore, the performances of these systems rely on the quality of the parsing results [14, 9]. To avoid these issues, deep-learning is used to develop systems learn high-level representations for each token, on which a classifier or sequence tagger can be trained [6].

Medical Document Anonymization (MEDDOCAN) [7] is a challenge in the shared task of IberLEf2019 dedicated to the EHRs in the Spanish language. There are two structured sub-tasks: "sensitive token detection" and "NER offset and entity type classification". The first sub-task aims to identify the sensitive tokens in a document. We can solve this sub-task as a token-level binary classification problem in which we develop a system that takes as input a document and classify each token as sensitive or not. The second sub-task aims at identifying the type of each token in a document. We can model this problem as a sequence tagging problem. The input is a sequence of tokens, and the output is their corresponding labels.

We participated in the MEDDOCAN challenge by developing E2EJ, a joint and end-to-end neural network-based system for the two sub-tasks. The proposed system provides an end-to-end solution and does not require any parsers or other linguistic resources. Specifically, the proposed system is a multilayer neural network, where the first three layers aim to learn high representation for a sequence of tokens, then we pass, jointly, the output of these layers to two submodels that are learned interactively. One is for extracting the sensitive tokens, while the other is for identifying their types.

The rest of the paper structured as follows: Section 2 presents the Methodology; Section 3 explains the dataset, baselines, and experimental settings; Section 4 presents and discusses the results; finally, Section 5 concludes this paper.

2 System Description

The main distinction point between our model and the literature deep-learning-based is the consideration of the interaction between the two tasks of sensitivity

detection and token type identification. In this subsection, we introduce E2EJ and its implementation steps in detail. Fig 1. depicts the architecture of our model.

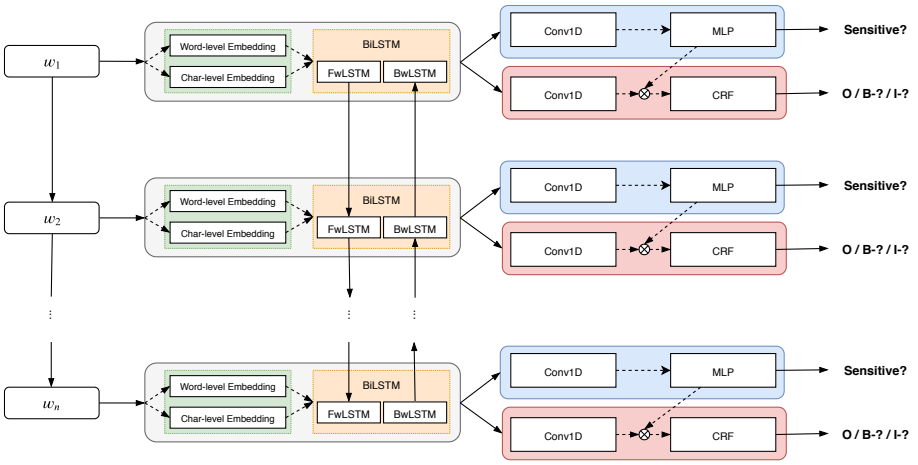

Fig. 1. E2EJ Architecture.

2.1 Embedding Layer

The goal of the embedding layer is to represent each word $w_i \in S$ by a low-dimensional vector space $v_i \in R^d$. Here, d is the size of the embedding layer. We use two levels of embedding: word-level and character-level. For the word-level embedding, we replace w_i with its pre-trained Glove word embedding vector v_i^w [11]. We use a single-layer 1-Dimensional Convolutional Neural Networks (Conv1D) with max-over-time pooling to represent the word at character-level as the following. Suppose that w_i is made up of a sequence of characters $[c_1, c_2, ..., c_n]$, where n is the length of w_i. First, we pass the sequence of characters of the word w_i to a randomly initialized character embedding layer to get the matrix $C_i \in R^{r \times l}$ - that is the character-level representation of w_i. Here, the $j-$th column corresponds to the character embedding for c_j. After that, we apply a narrow convolution between C_i and a filter (or kernel) $H \in R^{r \times k}$ of width k, after which we add a bias and apply a nonlinearity to obtain a feature map $f^i \in R^{n-k+1}$. Specifically, the m-th element of f^i is given by:

$$f^i[m] = tanh(\langle C_i[*, m : m + k - 1], H \rangle + b) \quad (1)$$

where $C_i[*, m : m + k - 1]$ is the m-to-$(m + w1)$-th column of C_i and $\langle A, B \rangle$ is the frobenius inner product. Finally, we take the max-over-time

$$v_i^c = max_m f^i[m] \quad (2)$$

as the feature corresponding to the filter H (when applied to word w_i).A filter is basically picking out a character n-gram, where the size of the n-gram corresponds to the filter width.

The final representation of the word w_i is given by concatenating the word-level vector and the character-level vector.

$$v_i = [v_i^w; v_i^c] \tag{3}$$

2.2 BiLSTM Layer

The goal of the encoder layer is to represent the sequence of words representations, $\{v_1, v_2, ..., v_l\}$, that is obtained from the embedding layer in higher level of abstraction and model the sequential phenomena. In this work we use a BiRNN to design our encoder. A BiRNN consists of forward $\overrightarrow{\phi}$ and backward $\overleftarrow{\phi}$ recurrent neural networks (RNNs). The first one reads the input sequence in a forward direction and produces a sequence of forward hidden states $(\overrightarrow{h_1}, ..., \overrightarrow{h_l})$, whereas the former reads the sequence in the reverse order $(v_{w_l}, ..., v_{w_1})$ resulting in a sequence of backward hidden states $(\overleftarrow{h_l}, ..., \overleftarrow{h_1})$.

We obtain a representation for each word v_{w_t} by concatenating the corresponding forward hidden state $\overrightarrow{h_t}$ and the backward one $\overleftarrow{h_t}$. The following equations illustrate the main ideas:

$$\overrightarrow{h_t} = \overrightarrow{\phi}(v_{w_t}, \overrightarrow{h_{t-1}}) \tag{4}$$

$$\overleftarrow{h_t} = \overleftarrow{\phi}(v_{w_t}, \overleftarrow{h_{t-1}}) \tag{5}$$

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \tag{6}$$

In practice, RNNs are challenging to train. Gradients may explode or vanish over long sequences [10]. To overcome these problems, we use Long Short-Term Memory (LSTM) [3] networks that are a more sophisticated variant of regular RNNs.

2.3 Sensitivity Detection Sub-Model

The input to this sub-model is the obtained sequence of vectors from the BiLSTM layer, and the output is the probability for each token been sensitive. As shown fig. 1, it comprises of two units: Conv1D with a single layer and a multi-layer perceptron (MLP) with one hidden layer and one Sigmoid neuron, i.e., the output layer. The goal of the Conv1D layer is to enrich the representation of each token with information about a fixed-sized context depending on a kernel width of k. Formally, we get the final representation of the input sequences as follows:

$$[v_1^s, v_2^s, ..., v_l^s] = Conv1D([v_1, v_2, ..., v_l] \tag{7}$$

Where $Conv1D$ refers to the same operations in Equations 1 and 2. Given that, for each v_t^s, we obtain the final output as the following.

$$x_t^s = tanh(v_t^s \cdot W_1^s + b_1^s) \tag{8}$$

$$\bar{y}_t^s = sigmoid(x_t^s \cdot W_2^s + b_2^s) \tag{9}$$

Here, $W_1^s \in R^{d_s \times d_x}$, $b_1^s \in R^{d_x}$, $W_2^s \in R^{d_s \times 1}$ and $b_2^s \in R$ are the MLP parameters. Where d_s is the dimensionality of the output vector from $Conv1D$ and d_x is the dimensionality of the output vector from the hidden layer.

2.4 NER Type Detection Sub-Model

Similarly, the input to this sub-model is the obtained sequence of vectors from the BiLSTM layer. The output, in this case, is the probability for each token been sensitive. Formally, let $[v_1^t, v_2^t, ..., v_l^t]$ be the sequence of vectors to be labeled, which is produced the concatenation of the MLP layer in the Sensitivity Detection sub-model and the output of the $Conv1D$ layer in this sub-model, and $Y^t = [y_1^t, y_2^t, ..., y_l^t]$ is the corresponding tag sequence. Each element y_i^t of y is one of the $B- < entity >$, $I- < entity >$ or O tags. Both H and Y^t are assumed to be random variables, and they are jointly modeled using a conditional random field (CRF).

2.5 Training

We train our model to minimise the joint objective function J.

$$J = J_s + J_t \tag{10}$$

Where J_s is the sigmoid cross-entropy and J_t is the negative log-probability of the correct tag sequence:

$$J_s = y_t^s \times -log(\bar{y}_t^s) + (1 - y_t^s) \times -log(1 - \bar{y}_t^s) \tag{11}$$

$$J_t = -log\left(p(Y^t|H)\right) \tag{12}$$

Where y_t^s is the golden label and $\bar{y}_t^s$ is the predicted one. The Y^t refers to the sequence of tags. As optimization algorithm, we used Stochastic Gradient Descent (SGD)-based ADAM algorithm [4] with learning-rate of 0.001. To avoid the over-fitting, we used dropout on the embeddings and decoder outputs with a rate of 0.3 [12].

3 Experiments

In this section, we discuss the dataset used and different experimental settings devised to evaluate our system.

3.1 Dataset Details

We trained and fine-tuned our system respectively on the training and the development sets provided by the organizers of the MEDDOCAN challenge. After that, we submitted the predicted labels of the test set that are produced by our system to evaluate its performance. The organizers omitted the golden labels of the test. The training set contains 500 documents, and the development and test sets contain 250 documents each.

3.2 Hyper-parameters

We used grid-search to obtain the best hyper-parameter values based on the development set. We list these values in Table 1.

Table 1. The chosen hyper-parameter values.

Word Embedding	Dimension size: 300 Initialization: Glove Trainable: No
Char Embedding	Dimension size: 50 Conv1D filters: 100 Kernel width: 3 Initialization: Uinform [-0.1, 0.1]
BiLSTM	Hidden units: 256 Layers: 2
Sub-Model (1)	Conv1D filters: 200 Kernel width: 3 Hidden size: 200
Sub-Model (2)	Conv1D filters: 200 Kernel width: 3

4 Results

We evaluated the performance of our system by comparing it against the following baseline systems:

- **RegEx**: a rule-based system using only regular expressions.
- **CRf**: a CRf-based system trained on a set of features such as unigram, part-of-tags, word shape, affixes, etc. [12]
- **E2E-LSTM**: a version of our system that are trained to only identify the type of tokens.

Table 2 shows the results of our submitted system (i.e., E2EJ system) and the compared systems. The evaluation metrics are precision, recall, and F1 scores. From the reported results, we can note that in general, E2EJ gives comparable performance to the state-of-the-art system CRF. It outperforms all the compared systems in terms of recall metric. One remarkable observation is that our system, unlike the other systems, gives a similar performance in all the evaluation metrics, which shows its consistency. Hence, some error analysis and performance inspection can lead to improving the performance of the system. The CRF-based system gives the best performance in terms of precision and F1 metrics with the NER sub-task, and the best performance in terms of the precision score for the Spans detection sub-task. We attribute this to the use of the external MEDDOCAN-Gazetteer resources provided by the organizers of the task.

Table 2. The Performances of our system compared to various methods. The best value in bold.

System	Sub-Task 1 (NER)			Sub-Task 2 (Spans)		
	P	R	F1	P	R	F1
RegEx	91.06	81.01	85.74	91.32	81.24	85.99
CRf	**97.02**	94.93	**95.96**	**97.47**	95.37	96.41
E2E	94.78	93.64	94.21	95.80	94.65	95.22
E2EJ	95.98	**95.69**	95.83	96.76	**96.45**	**96.61**

5 Conclusion

We have developed a system, called E2EJ, that automatically detects the sensitive entities and identify their types in Spanish electronic health records. It contains two sub-models that are trained jointly. The first one aims to detects the sensitive entities and guides the second one to accurately predict the type of these detected tokens. E2EJ provides an end-to-end solution and does not require any external tools or other linguistic resources. The effectiveness of the proposed system has been evaluated by participating in the Medical Document Anonymization challenge for the electronic health records in Spanish language obtaining results which show comparable results to the state-of-the-art systems and outperform the baseline systems. The reported results show that the proposed system is stable and consistent. In our future work, we plan to perform extensive error analysis and inspect the performance of the system and improve it. For example, we plan to use a transformer-based interpretable model like BERT [2] as a pre-trained sentence encoder instead of using BiLSTM.

Acknowledgements

The authors acknowledge the support of Univ. Rovira i Virgili through a Martí i Franqués PhD grant, the assistant/teaching grant for the Department of Computer Engineering and Mathematics and the Research Support Funds 2019PFR-URV-B2-60.

References

1. Act, A.: Health insurance portability and accountability act of 1996. Public law **104**, 191 (1996)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
3. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
4. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

5. Liu, Z., Chen, Y., Tang, B., Wang, X., Chen, Q., Li, H., Wang, J., Deng, Q., Zhu, S.: Automatic de-identification of electronic medical records using token-level and character-level conditional random fields. Journal of biomedical informatics **58**, S47–S52 (2015)
6. Liu, Z., Tang, B., Wang, X., Chen, Q.: De-identification of clinical notes via recurrent neural network and conditional random field. Journal of biomedical informatics **75**, S34–S42 (2017)
7. Marimon, M., Gonzalez-Agirre, A., Intxaurrondo, A., Rodrguez, H., Lopez Martin, J.A., Villegas, M., Krallinger, M.: Automatic de-identification of medical texts in spanish: the meddocan track, corpus, guidelines, methods and evaluation of results. In: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019). vol. TBA, p. TBA. CEUR Workshop Proceedings (CEUR-WS.org), Bilbao, Spain (Sep 2019), TBA
8. Meystre, S.M., Friedlin, F.J., South, B.R., Shen, S., Samore, M.H.: Automatic de-identification of textual documents in the electronic health record: a review of recent research. BMC medical research methodology **10**(1), 70 (2010)
9. Neamatullah, I., Douglass, M.M., Li-wei, H.L., Reisner, A., Villarroel, M., Long, W.J., Szolovits, P., Moody, G.B., Mark, R.G., Clifford, G.D.: Automated de-identification of free-text medical records. BMC medical informatics and decision making **8**(1), 32 (2008)
10. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: ICML (2013)
11. 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)
12. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research **15**(1), 1929–1958 (2014)
13. Sweeney, L.: Replacing personally-identifying information in medical records, the scrub system. In: Proceedings of the AMIA annual fall symposium. p. 333. American Medical Informatics Association (1996)
14. Uzuner, Ö., Luo, Y., Szolovits, P.: Evaluating the state-of-the-art in automatic de-identification. Journal of the American Medical Informatics Association **14**(5), 550–563 (2007)