NLNDE: The Neither-Language-Nor-Domain-Experts' Way of Spanish Medical Document De-Identification

Lukas Lange[1,2,3], Heike Adel[1], and Jannik Strötgen[1]

[1] Bosch Center for Artificial Intelligence
Robert-Bosch-Campus 1, 71272 Renningen, Germany
{Lukas.Lange,Heike.Adel,Jannik.Stroetgen}@de.bosch.com
https://www.bosch-ai.com
[2] Spoken Language Systems (LSV),
[3] Saarbrücken Graduate School of Computer Science
Saarland Informatics Campus, Saarland University, Saarbrücken, Germany

Abstract. Natural language processing has huge potential in the medical domain which recently led to a lot of research in this field. However, a prerequisite of secure processing of medical documents, e.g., patient notes and clinical trials, is the proper de-identification of privacy-sensitive information. In this paper, we describe our NLNDE system, with which we participated in the MEDDOCAN competition, the medical document anonymization task of IberLEF 2019. We address the task of detecting and classifying protected health information from Spanish data as a sequence-labeling problem and investigate different embedding methods for our neural network. Despite dealing in a non-standard language and domain setting, the NLNDE system achieves promising results in the competition.

Keywords: De-Identification · Recurrent Neural Networks · Embeddings

1 Introduction

The anonymization of privacy-sensitive information is of increasing importance in the age of digitalization and machine learning. It is, in particular, relevant for texts from the medical domain that contain a large number of sensitive information by nature. The shared task MEDDOCAN (Medical Document Anonymization) [11] aims at automatically detecting protected health information (PHI) from Spanish medical documents. Following the past de-identification task on English PubMed abstracts [14], it is the first competition on this topic on Spanish data.

In this paper, we describe our submissions to MEDDOCAN and their results. We, as **N**either **L**anguage **N**or **D**omain **E**xperts (NLNDE), address the anonymization task as a sequence-labeling problem and use a combination of different state-of-the-art approaches from natural language processing to tackle its challenges.

We train recurrent neural networks with conditional random field output layers which are state of the art for different sequence labeling tasks, such as named entity recognition [9], part-of-speech tagging [7] or de-identification [8,10]. Recently, the field of natural language processing has seen another boost in performance by using context-aware language representations which are pre-trained on a large amount of unlabeled corpora [1,4,12]. Therefore, we experiment with FLAIR embeddings for Spanish [1] to represent the input of our networks. In our different runs, we further explore the advantages of domain-specific fastText embeddings [3] that have been pre-trained on SciELO and Wikipedia articles [13].

From a natural-language-processing perspective, the MEDDOCAN task is interesting due to the non-standard domain (medicine) and language (Spanish) of the documents. The results of our submissions show that state-of-the-art architectures for sequence-labeling tasks can be directly transferred to these settings and that domain-specific embeddings are helpful but not necessary.

2 Methods

In this section, we first give an overview of the different embedding methods we use in our system. Second, we describe the architecture of our system.

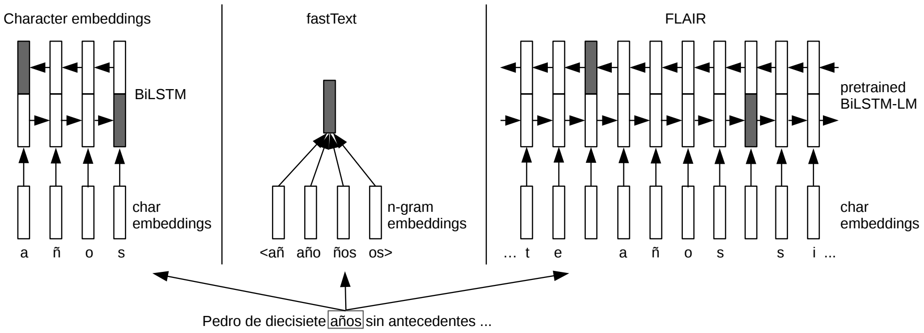

Fig. 1. Comparison of sub-word embedding methods. The shaded vectors are used to represent the input token.

2.1 Sub-word Embedding Methods

In our different runs, we investigate the impact of the following sub-word embedding methods: character-based, fastText and FLAIR embeddings. They are depicted in Figure 1.

Character Embedding: The characters of a word are represented by randomly initialized embeddings. Those are passed to a bi-directional long short-term memory network (BiLSTM). The last hidden states of the forward and backward pass are concatenated to represent the word [9].

FastText Embedding: The fastText embeddings represent a word by the normalized sum of the embeddings for the n-grams of the word [3].

FLAIR Embedding: FLAIR computes character-based embeddings for each word depending on all words in the context [2]. For this, the complete sentence is used as the input to the BiLSTM instead of only a single word. The BiLSTM of FLAIR is pretrained using a character-level language model objective, i.e., given a sequence of characters, compute the probability for the next character.

2.2 NLNDE System

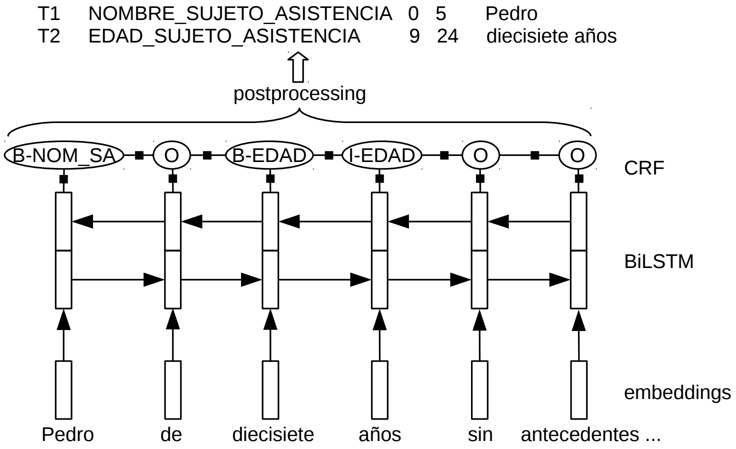

Fig. 2. General architecture of all our models. The label prefixes "B-" and "I-" show how we address the task as a sequence-labeling task.

In Figure 2, the architecture of our model is depicted. In the following, we explain the different layers.

Input Representation. We tokenize the input using the tokenizer provided by the shared task organizers [6]. Then, we represent each token with embeddings. In our runs, we investigate the impact of the following kinds of embeddings: the output of an LSTM over character embeddings (50 dimensions, randomly initialized and fine-tuned during training), domain-independent fastText embeddings (300 dimensions, pre-trained on Spanish text [5]), domain-specific fastText embeddings (100 dimensions, pre-trained on Spanish SciELO and Wikipedia articles [13]), and FLAIR embeddings (4096 dimensions, pre-trained on Spanish text [2]). For FLAIR embeddings, we also test their pooled version (8192 dimensions, using min pooling) [1]. Note that except for the character embeddings, we do not fine-tune any of the embeddings.

BiLSTM-CRF Layers. The embeddings are fed into a BiLSTM with a conditional random field (CRF) output layer, similar as done by Lample et al. [9]. The CRF output layer is a linear-chain CRF, i.e., it learns transition scores between the output classes. For training, the forward algorithm is used to sum the scores for all possible sequences. During decoding, the Viterbi algorithm is applied to obtain the sequence with the maximum score. Note that the hyperparameters are the same across all runs. We use a BiLSTM hidden size of 256 and train the network with mini-batch stochastic gradient descent using a learning rate of 0.1 and a batch size of 32. For regularization, we employ early stopping on the development set and apply dropout with probability 0.5 on the input representations.

Postprocessing. The output of the model is further adjusted with a post-processing layer, similar as done by Yang et al. [15] and Liu et al. [10]. As some classes from the annotation guidelines [4] do not occur in the training data, we tackle them with pattern matching. For this, we use regular expressions for URLs, IP- and MAC addresses to detect the classes `URL_WEB` and `DIREC_PROT_INTERNET`, overwriting the results of the neural classifier.

3 Submissions

We submitted five runs to the MEDDOCAN competition. All of them are based on the architecture described in Section 2.2. They only differ in the usage of different input representations.

S1 (*Char+fastText+Domain*): Our first run uses a combination of character embeddings, domain-independent fastText embeddings as well as domain-specific fastText embeddings to represent tokens. The resulting representation for each token has 450 dimensions.

S2 (*FLAIR+fastText*): In contrast to all other runs, the second run uses only domain-independent embeddings, i.e., embeddings that have been trained on standard narrative and news data from Common Crawl and Wikipedia. In particular, it uses a combination of domain-independent fastText embeddings and Flair embeddings.

S3 (*FLAIR+fastText+Domain*): The third run adds domain-specific fastText embeddings to the system of the second run in order to investigate the impact of domain knowledge.

[4] http://temu.bsc.es/meddocan/index.php/annotation-guidelines/

S4 (*PooledFLAIR*): The fourth run is equal to the third run, except that we use the minimum-pooling version of the FLAIR embeddings.

S5 (*Ensemble*): The fifth run is an ensemble of the previous four runs using weighted voting: Each classifier C_i is assigned a weight $w_i \in [0.5, 3]$. For each output label, the weights of the classifiers predicting it are summed. Then, the label with the highest score is chosen if it exceeds a specific threshold $t \in [1, 5]$, or O (no PHI class) otherwise. The weights and threshold are selected based on results on the development set as follows: $w_1 = 0.5$, $w_2 = 2.0$, $w_3 = 2.5$, $w_4 = 0.5$ and $t = 3$. With these parameters, a label needs votes from at least two classifiers ($w_i < t, i \in \{1, 2, 3, 4\}$). However, the models of the submissions S2 and S3 are assigned higher weights than S1 and S4. This reflects their performance (see next section).

4 Results and Analysis

This section describes our results and analysis. We report the results on the MEDDOCAN test set using the official shared task evaluation measures [11].

4.1 Results for Task 1: NER Offset and Entity Type Classification

In the first sub-task, the systems need to find spans for de-identification and categorize them into one of 29 classes. Table 1 presents our results on this subtask.

Table 1. Results of our five runs for Task 1.

$\mathbf{S}_{ID}$	**Leak**	**Precision**	**Recall**	**F1**
S1 (*Char+fastText+Domain*)	0.02432	0.96956	0.96767	0.96861
S2 (*FLAIR+fastText*)	0.02378	**0.97078**	0.96838	0.96958
S3 (*FLAIR+fastText+Domain*)	**0.02299**	0.96978	**0.96944**	**0.96961**
S4 (*PooledFLAIR*)	0.02724	0.96720	0.96379	0.96549
S5 (*Ensemble*)	0.02365	0.97044	0.96856	0.96950

While the domain-independent system (run 2 with FLAIR and domain-independent fastText embeddings) leads to the highest recall values, the third run that also uses domain-specific fastText embeddings achieves the highest F1 scores. This shows that integrating domain knowledge into the token representation is beneficial. However, the differences among the five runs are rather small, indicating that the architecture itself is already strong enough for the given dataset and the impact of different input representations is minor.

4.2 Results for Task 2: Sensitive Token Detection

Tables 2 and 3 provide the results of our models on the second sub-task (sensitive token detection). In contrast to task 1, this is a binary classification task.

Since the official evaluation measure for this task is the strict one, we focus our explanation on Table 2. The main ranking of our models is the same as the ranking for sub-task 1: the addition of domain-specific input representations performs best. Interestingly, the domain-specific input representations (run 3) now perform best in terms of recall as well while the domain-independent input representations (run 2) perform best in terms of precision.

Table 2. Results of our five runs for Task 2 (Evaluation: Strict).

$\mathbf{S}_{ID}$	**Precision**	**Recall**	**F1**
S1 (*Char+fastText+Domain*)	0.97522	0.97333	0.97427
S2 (*FLAIR+fastText*)	**0.97574**	0.97333	0.97453
S3 (*FLAIR+fastText+Domain*)	0.97508	**0.97474**	**0.97491**
S4 (*PooledFLAIR*)	0.97217	0.96873	0.97045
S5 (*Ensemble*)	0.97540	0.97350	0.97445

Table 3. Results of our five runs for Task 2 (Evaluation: Merged).

$\mathbf{S}_{ID}$	**Precision**	**Recall**	**F1**
S1 (*Char+fastText+Domain*)	**0.98749**	**0.98311**	**0.9853**
S2 (*FLAIR+fastText*)	0.98648	0.98145	0.98396
S3 (*FLAIR+fastText+Domain*)	0.98566	0.98264	0.98415
S4 (*PooledFLAIR*)	0.98182	0.97730	0.97956
S5 (*Ensemble*)	0.98598	0.98162	0.98380

In both sub-tasks, FLAIR embeddings outperform standard character embeddings (except for the evaluation type merge in Table 3). Also, for both sub-tasks, pooling of FLAIR embeddings leads to worse results. Surprisingly, run 5, i.e., the ensemble of the models from runs 1–4, does not improve the results over single models.

4.3 Confusion Matrix Analysis

Table 4 shows the confusion matrix of our best performing system (run 3). It is similar to the identity matrix, i.e., confusions between classes happen very rarely. The most confusions happen with O, the label we assign to all non-PHI terms which might be caused by the high number of occurrences of this class in the training dataset. Confusions among PHI-classes happen mostly between related classes. For example, Hospital (HOS) and Institution (INST) are confused quite often, as Hospital is a subclass of Institution and other medical institutions are tagged with Hospital and vice versa, e.g., *Clinica Gnation* is an institution

[2] Abbreviations for entity types:
CALLE (CALLE), CENTRO_SALUD (CS), CORREO_ELECTRONICO (MAIL), EDAD_SUJETO_ASISTENCIA (EDAD), FAMILIARES_SUJETO_ASISTENCIA (FAM), FECHAS (FECHA), HOSPITAL (HOS), ID_ASEGURAMIENTO (ID_AS), ID_CONTACTO_ASISTENCIAL (ID_CON), ID_EMPLEO_PERSONAL_SANITARIO (ID_EPS), ID_SUJETO_ASISTENCIA (ID_SA), ID_TITULACION_PERSONAL_SANITARIO (ID_TPS), INSTITUCION (INST), NOMBRE_PERSONAL_SANITARIO (NOM_PS), NOMBRE_SUJETO_ASISTENCIA (NOM_SA), NUMERO_FAX (#FAX), NUMERO_TELEFONO (#TEL), OTROS_SUJETO_ASISTENCIA (OTRO), PAIS (PAIS), PROFESION (PROF), SEXO_SUJETO_ASISTENCIA (SEXO), TERRITORIO (TER)

Table 4. Confusion matrix of the best model (S3) on the development set.[2]

Gold Label \ Predicted Label	O	CALLE	CS	MAIL	EDAD	FAM	FECHA	HOS	ID_AS	ID_CON	ID_EPS	ID_SUJ	ID_TPS	INST	NOM_PS	NOM_SA	#FAX	#TEL	OTRO	PAIS	PROF	SEXO	TER
O	123293	21			2	23	9	3		1		4	6	28	3		1			2		3	
CALLE	15	2997						2						1	8	1							6
CS			8					3															
MAIL				256											3								
EDAD	14				1014	3																	
FAM	18				2	104						2							2				
FECHA	16						1089																
HOS	10	4					4	551						11									1
ID_AS									573			2	6										
ID_CON										32													
ID_EPS	4																						
ID_SA	9											293									2		
ID_TPS													663									1	
INST	35	8	3					11						190									
NOM_PS	3					1									1585	2							
NOM_SA	3															779							
#FAX																	16						
#TEL	1																	70					1
OTRO	11					2						1							2				
PAIS																				349			
PROF	8																				1		
SEXO																						456	
TER	9	20					1	5						5						3			1141

tagged as a hospital. Analogously, Streets (CALLE) and Territoriums (TER) are getting confused often, as both classes are related and typically constitute of multiple tokens. In contrast to this, Countries (PAIS) are tagged correctly almost every time, as there is only a very limited number of countries and they are usually single token expressions.

4.4 Synthetic Augmentation Case Study

As mentioned above, the performance difference between our systems is rather small. This may be caused by the synthetic augmentation of the MEDDOCAN data which was used to extend the texts with header and footer information containing many PHI terms. In fact, 85% of PHI terms appear in the augmented text parts. While this extension is necessary to cover more classes and PHI terms, the synthetic nature of these extensions may have an impact on the performance of automatic classifiers. Therefore, we perform a case study in which we remove these parts from the test set and compare only the predictions found in the real text. Only 838 out of 5661 (14.8%) annotations and only 13 out of 29 classes remain in this experiment. The performances of our systems are decreased to F1 scores around 0.90 which is still rather high. This shows that our systems have learned more than just to reproduce the synthetic data augmentation. However, the performance differences among our systems are still small, indicating that the data augmentation was not the reason for this behavior. Note, however, that we did not retrain our models without the synthetic augmentation.

5 Conclusions

In this paper, we described the system with which we participated in the MEDDOCAN competition on automatically detecting protected health information from Spanish medical documents. As neither language nor domain experts, we addressed the task with a sequence labeling model. In particular, we trained a bi-directional long short-term memory network and explored different input representations. All of our runs achieved high performance with F1 scores about 97%.

References

1. Akbik, A., Bergmann, T., Vollgraf, R.: Pooled contextualized embeddings for named entity recognition. In: Proc. of NAACL. pp. 724–728 (2019)
2. Akbik, A., Blythe, D., Vollgraf, R.: Contextual string embeddings for sequence labeling. In: Proc. of COLING. pp. 1638–1649 (2018)
3. 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). https://doi.org/10.1162/tacl_a_00051
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proc. of NAACL. pp. 4171–4186 (2019)
5. Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T.: Learning word vectors for 157 languages. In: Proc. of LREC (2018)
6. Intxaurrondo, A.: SPACCC (spanish clinical case corpus) tokenizer (Mar 2019). https://doi.org/10.5281/zenodo.2586978
7. Kemos, A., Adel, H., Schütze, H.: Neural semi-Markov conditional random fields for robust character-based part-of-speech tagging. In: Proc. of NAACL. pp. 2736–2743 (2019)
8. Khin, K., Burckhardt, P., Padman, R.: A deep learning architecture for de-identification of patient notes: Implementation and evaluation. CoRR **abs/1810.01570** (2018)
9. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: Proc. of NAACL (2016)
10. 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). https://doi.org/https://doi.org/10.1016/j.jbi.2017.05.023
11. 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)
12. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proc. of NAACL (2018)
13. Soares, F., Villegas, M., Gonzalez-Agirre, A., Krallinger, M., Armengol-Estapé, J.: Medical word embeddings for Spanish: Development and evaluation (Jun 2019), https://www.aclweb.org/anthology/W19-1916
14. Stubbs, A., Uzuner, Ö.: Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/uthealth corpus. Journal of biomedical informatics **58**, S20–S29 (2015)
15. Yang, H., Garibaldi, J.M.: Automatic detection of protected health information from clinic narratives. Journal of Biomedical Informatics **58**, S30 – S38 (2015). https://doi.org/https://doi.org/10.1016/j.jbi.2015.06.015