Vicomtech at MEDDOCAN: Medical Document Anonymization

Naiara Perez*, Laura García-Sardiña*, Manex Serras*, and Arantza Del Pozo

Vicomtech, Mikeletegi Pasealekua, 57, 20009 - Donostia/San Sebastián, Spain
{nperez,lgarcias,mserras,adelpozo}@vicomtech.org

Abstract. This paper describes the participation of Vicomtech's team in the *MEDDOCAN: Medical Document Anonymization* challenge, which consisted in the recognition and classification of protected health information (PHI) in medical documents in Spanish. We tested different state-of-the-art classification algorithms, both deep and shallow, and rich sets of features, obtaining an F1-score of 0.960 in the strictest evaluation. The models submitted and scripts for decoding will be available at https://snlt.vicomtech.org/meddocan2019.

Keywords: PHI De-identification · Textual Anonymisation · Machine Learning · Spanish Corpus

1 Introduction

The major bottleneck for the advancement of Natural Language Processing (NLP) in the medical field is the struggle in accessing real clinical texts, mainly due to data privacy protection issues. *MEDDOCAN: Medical Document Anonymization* [9] is the first challenge devoted to the recognition and classification of protected health information (PHI) in medical documents in Spanish. The challenge has two sub-tasks: NER offset and entity type classification, and sensitive span detection.

This paper describes the participation of Vicomtech's team in the MEDDOCAN challenge. Our aim has been to test a variety of state-of-the-art approaches, whether neural or shallow, as well as their combinations. Specifically, Conditional Random Fields (CRF) [7] are prominently featured, having been extensively used for tasks of sequential nature such as named entity recognition [10] and for textual sensitive data identification and anonymisation [4,3]; other techniques used include XGBoost [2], Convolutional Neural Networks (CNN) [8], and Long Short-Term Memories (LSTM) [6]. The models submitted and auxiliary scripts for feature extraction and decoding will be freely available at https://snlt.vicomtech.org/meddocan2019.

The paper is structured as follows: Section 2 starts describing the task's data and the set of features extracted; then, the systems are presented, with a focus on the practicalities of the implementations than on theoretical explanations. The

* Contributed equally

results obtained are reported in Section 3 and discussed in Section 4. Finally, the paper ends by presenting the conclusions and hints for future work in Section 5.

2 Materials and Methods

2.1 Data

The MEDDOCAN corpus consists of clinical cases written in Spanish and manually enriched with PHI expressions. A total number of 22 PHI categories are considered which show high frequency variability[1]. The pre-processing and formatting applied to the corpus consisted of the following steps:

1. **Paragraph splitting.** Documents were split into paragraphs using line breaks in the original texts. We decided to work with paragraphs instead of sentences because the provided sentence-splitting tool occasionally split parts of target entities into different sentences.
2. **Tokenisation.** Each paragraph was tokenised using the SPACCC Part-of-Speech Tagger[2] and some extra custom tokenisation rules, mainly to split punctuation symbols if not inside a URL, email address or date, and to split camel cased words in order to account for spacing errors in the original text (e.g., 'DominguezCorreo' into 'Dominguez Correo').
3. **Label formatting.** The Brat-formatted annotations of the training and development datasets were converted to token level tags following the BILOU (*Beginning*, *Inner*, *Last*, *Outside*, *Unique*) scheme. Combining this tag scheme with the original 22 granular PHI classes (e.g., for the granular class `FECHA` we would have the tags `B-FECHA`, `I-FECHA`, `L-FECHA`, `U-FECHA`, plus the generic `O` class) gives a final tag set of 89 possible unique labels.

The final statistics including the number of documents, paragraphs, tokens, vocabulary size, and PHI entities for each of the datasets in the pre-processed corpus can be consulted in Table 1.

Table 1. Final statistics of the pre-processed datasets

	# docs	# pars	# toks	vocab	# PHI
train	500	10,311	260,407	26,355	11,333
development	250	5,268	138,812	15,985	5,801
test	250	5,155	132,961	15,397	5,661

[1] The MEDDOCAN annotation scheme defines 29 PHI entity types, but only 22 of them actually appear in the annotated sets.

[2] https://github.com/PlanTL-SANIDAD/SPACCC_POS-TAGGER

2.2 Features

The complete set of features extracted to train the classifiers is listed succinctly in Table 2. Note, however, that the final submitted results were obtained drawing upon a different set of features in each case (detailed in Section 2.3).

2.3 Systems

Our team submitted 5 systems' results to the MEDDOCAN task. The same systems were used for both sub-tasks: *i)* NER offset and entity type classification, and *ii)* sensitive span detection. All the systems were complemented with a small set of rules that annotated numerical expressions, such as dates and phone numbers, via regular expressions. However, these rules had little impact on the final results. Next, predicted labels were post-processed to ensure that the result followed the BILOU scheme, having the BILOU tag prevail over the PHI category tag (e.g., the sequence `B-CALLE > L-PROFESION` would be converted to `B-CALLE > L-CALLE` instead of `U-CALLE > U-PROFESION`). Finally, the predictions had to be converted back to Brat's format.

spaCy As a first approach to the task, we experimented with spaCy's[3] Named Entity Recogniser (NER), built on Bloom Embeddings [13] and residual Convolutional Neural Networks [5]. We followed the given recipe [4] with default settings and applied the recommended tweaks: compounding batch size, dropout decay, and parameter averaging.

spaCy supports a closed set of features, which overlaps only partially with our own. Interestingly, training an empty model yielded better results on the development set than using the accepted features. Likewise, training embeddings from scratch also gave better results than using those presented in Section 2.2 as pre-trained embeddings. Thus, the results submitted to the task were obtained with a NER model trained from scratch, with no extra information provided but the training data.

CRF The second run presented corresponded to a system based on Conditional Random Fields, implemented using the python sklearn-crfsuite library. The final CRF model did not include word embeddings or date-time expressions as features, because they provided slightly worse results in previous feature selection trials explored to reduce dimensionality. Features with float values were rounded to one decimal. The final system was trained using the configuration presented in Table 3.

[3] https://spacy.io
[4] https://spacy.io/usage/training#ner

Table 2. Set of features

Token characterisation

Token: the token itself.
Length: the length in characters of the token.
Casing: features related to the token's casing, i.e., whether the token is uppercase, lowercase or titlecase, and the ratio of uppercase characters to the token's length.
Digits and punctuation: features related to the token's character types, e.g., whether the token is alphanumeric or a punctuation mark, the ratio of the number of punctuation marks to the token's length, and so on.
Affixes: the token's first and last character bigrams and trigrams.

Term characterisation

Linguistic information: the lemma and part-of-speech tag given by the SPACCC tagger at the data pre-processing step.
NERC: the named entity tag given by spaCy's model es_core_news_md 2.1.0. If a detected named entity was multi-word, we gave the same tag to all the tokens involved.
Date-time expressions: whether the token is part of a date and/or time expression according to a left-to-right parser designed beforehand.
Gazetteers: the maximum similarity score obtained when matching text n-grams with gazetteer entries. We used a total of 10 gazetteers: the ones provided by the organisers[*], plus country names, kinship relations, months, and sexes. The string similarity was computed with the python-Levenshtein library and was only added as feature if it was greater than 0.75. If a match was multi-word, we gave the same score to all the tokens involved.
Brown clusters: complete paths and paths pruned at lengths 8, 16, 32, and 64. The clusters [1] were computed on the training set's vocabulary with tan-clustering[**], using the default settings of the tool.
Word vectors: each dimension in the word vectors provided by the task organisers [15]. Specifically, we used the Word2Vec embeddings [11,12] of 300 dimensions trained on SciELO and Wikipedia.

Context characterisation

Boundaries: whether the token is first or last in the paragraph.
Length: the length in tokens of the paragraph the token belongs to.
Position: the position of the paragraph in the document.
Header: the nearest expression to the left of each token that is followed by a colon, lowercased (e.g., 'email:', 'antecedentes familiares:', and so on).
Window: all the features of the neighbouring tokens in a ±3 context window with respect to the current token, except for the paragraph length and position.

[*] http://temu.bsc.es/meddocan/index.php/resources/
[**] https://github.com/mheilman/tan-clustering

Table 3. CRF configuration

parameter	value	parameter	value	parameter	value
algorithm	`lbfgs`	c1	`0.1`	all transitions	`True`
maximum iterations	`100`	c2	`0.1`		

CRF-XGBoost ensemble The third run corresponds to a system that combines the non-sequential output of multiple XGBoost classifiers. The XGBoost algorithm has achieved top-tier results in multiple Kaggle competitions of different characteristics.

The first layer of the system is built with multiple XGBoost models trained using different tagging schemes in addition to BILOU: *i)* 1/0 considering if the token is PHI or not; *ii)* the token's PHI granular class, without considering any sequence tagging scheme (i.e., `CALLE` instead of `B-CALLE`); and *iii)* BIO, a more relaxed version of BILOU considering *Beginning*, *Inner*, *Outside* positions.

To train these models, the full set of features described in Section 2.2 was reduced to the top 200k features according to the univariate statistical test ANOVA [14]. Finally, a CRF classifier with a ±5 window was trained to tackle the lack of context of the XGBoost algorithm using the different models' predictions and the token, obtaining a more sequentially coherent result.

NCRF++ NCRF++ [16] is an open-source toolkit built on PyTorch to train neural sequence labelling models. We kept the default network configuration[5]: 4 CNN layers for character sequence representations, a bidirectional LSTM layer for word sequence representations and an output CRF layer. The hyperparameter settings are shown in Table 4. Regarding the features, in this case we used all the available ones except for those derived from the word vectors, as these were given as pre-trained embeddings to the network.

Table 4. NCRF++ hyperparameters

parameter	value	parameter	value	parameter	value
epochs	`30`	char emb dim	`30`	learning rate	`0.01`
batch size	`100`	char hidden dim	`50`	L_2 regularisation	`1e-6`
optimiser	`Adam`	word emb dim	`300`		
average batch loss	`True`	word hidden dim	`150`		

[5] The maximum sentence length was hard-coded to 250 tokens at training; this threshold had to be removed in prediction to accommodate a few longer sentences.

Weighted Voting This run was an ensemble of the previous four taggers, where each tagger's vote was given a weight equal to the F1-score obtained on the development set in order to resolve potential ties.

3 Results

Table 5 shows the results achieved for both sub-tasks. Results are given in terms of micro-averaged precision, recall, and F1-score. In addition, a leak score is reported for the first sub-task. The leak score is the ratio of false negatives to the number of sentences.

Table 5. Final results on the development and test sets

	development set				test set			
	P	R	F1	leak	P	R	F1	leak
NER offset and entity type classification sub-task								
spaCy	0.968	0.943	0.955	0.043	0.965	0.948	0.956	0.039
CRF	0.969	9.929	0.948	0.077	**0.971**	0.937	0.954	0.048
CRF-XGB	0.965	0.942	0.953	0.043	0.966	0.945	0.955	0.041
NCRF++	0.964	0.947	0.956	0.039	0.964	**0.956**	**0.960**	**0.032**
W. Voting	**0.975**	**0.953**	**0.964**	**0.035**	0.968	0.947	0.958	0.039
Sensitive span detection sub-task								
spaCy	0.966	0.945	0.955	-	0.967	0.953	0.960	-
CRF	0.974	0.933	0.953	-	**0.977**	0.943	0.960	-
CRF-XGB	0.968	0.945	0.957	-	0.971	0.950	0.960	-
NCRF++	0.970	0.953	0.962	-	0.972	**0.964**	**0.968**	-
W. Voting	**0.978**	**0.956**	**0.967**	-	0.975	0.954	0.964	-

All the systems achieved F1-scores over 0.950 on the test set, the best F1-scores being 0.960 and 0.968 for the first and second sub-tasks, respectively. All systems favour precision over recall. Among individual systems, NCRF++ has the best scores; particularly, it has a markedly better recall than the rest. On the other hand, CRF outperforms the other systems in terms of precision, but the lower recall relegates it to the last position in the rank. Weighted Voting improves notably every individual score on the development set, but does not prove to be that helpful on the test set. On the contrary, individual systems show slightly better results on the test set than on the development set. This improvement is more pronounced for recall. As for the CRF-XGBoost ensemble and spaCy, they perform quite similar and remain ranked third and fourth in both sub-tasks.

4 Discussion

Although our focus is set on the final systems, it is worth mentioning that several non-final versions were trained on data labelled using different tagging schemes. Results of these models on the development set showed that using the BILOU tagging scheme outperformed using other labelling practices.

We also run some trials using models trained specifically for the second subtask, i.e., without using the PHI granular class labels. However, results on the development set usually showed no significant improvement over models trained using PHI class information. For this reason, the same final classifiers were used for both sub-tasks.

Despite the vast amount of features extracted for the task, different feature selection algorithms determined that the most relevant ones were Brown clusters, followed by affixes. Context characterisation features were also denoted as relevant, partly influenced by the semi-structured nature of the documents.

A tentative error analysis showed that the systems made very similar errors, although with varying frequencies. Most of the false negatives involved entities located at the least structured parts of the documents and usually consisted of mentions to the patients' relatives, professions, and other types of less frequent PHI categories. Another type of PHI class difficult to predict correctly was addresses, because the systems segmented them into spans different to those in the gold annotations. Finally, phone, fax, and identification numbers were correctly recognised but incorrectly categorised on a few occasions. Regarding false positives, most of them corresponded to improperly segmented addresses and the misclassification of numeric expressions. The rest of falsely predicted PHI items were most frequently entities seemingly missed by the human annotators.

5 Conclusions and Future Work

In this paper we described Vicomtech's approach to the MEDDOCAN challenge, which consisted in trying different state-of-the-art Machine Learning classification algorithms, both deep and shallow, and rich sets of features. Such approach proved to be effective, since all of the five final submitted systems achieved F1-scores over 0.95 and 0.96 on the test set for the first and second sub-tasks respectively. The final models submitted and auxiliary scripts for decoding will be freely available at https://snlt.vicomtech.org/meddocan2019.

Best results were achieved by a neural sequence classifier, followed by a weighted voting ensemble system. Still, the results of other participants are unknown to us at the time of writing and, thus, no conclusions can be reached as to the competitiveness of the presented systems.

Future work includes a deeper error analysis, in order to elucidate the differences in the results obtained on the development and test sets, and the proposal of solutions to approach recurrent classification errors. For example, spotted errors regarding family kinship or race could probably be solved with simple post-processing dictionary look-up heuristics. An analysis of which features turn out more and less important for the classifiers' learning could also provide relevant information for building new systems to tackle similar tasks.

References

1. Brown, P.F., Desouza, P.V., Mercer, R.L., Pietra, V.J.D., Lai, J.C.: Class-based n-gram models of natural language. Computational linguistics **18**(4), 467–479 (1992)
2. Chen, T., Guestrin, C.: XGBoost: A Scalable Tree Boosting System. In: Proc. of ACM SIGKDD 2016. pp. 785–794 (2016)
3. García-Sardiña, L., Serras, M., del Pozo, A.: Knowledge transfer for active learning in textual anonymisation. In: Proc. of SLSP 2018. pp. 155–166 (2018)
4. He, B., Guan, Y., Cheng, J., Cen, K., Hua, W.: CRFs based de-identification of medical records. Journal of Biomedical Informatics **58**, S39–S46 (2015)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Proc. of IEEE CVPR 2016. pp. 770–778 (2016)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997)
7. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proc. of ICML 2001. pp. 282–289 (2001)
8. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Computation **1**(4), 541–551 (1989)
9. 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: Proc. of IberLEF 2019. p. TBA (2019)
10. McCallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: Proc. of HLT NAACL 2003. pp. 188–191 (2003)
11. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proc. of NIPS 2013. pp. 3111–3119 (2013)
12. Mikolov, T., Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proc. of NAACL HLT 2013. pp. 746–751 (2013)
13. Serrà, J., Karatzoglou, A.: Getting Deep Recommenders Fit: Bloom Embeddings for Sparse Binary Input/Output Networks. In: Proc. of RecSys 2017. pp. 279–287 (2017)
14. Sheikhan, M., Bejani, M., Gharavian, D.: Modular neural-SVM scheme for speech emotion recognition using ANOVA feature selection method. Neural Computing and Applications **23**(1), 215–227 (2013)
15. Soares, F., Villegas, M., Gonzalez-Agirre, A., Krallinger, M., Armengol-Estapé, J.: Medical word embeddings for Spanish: Development and evaluation. In: Proc. of Clinical NLP Workshop 2019. pp. 124–133 (2019)
16. Yang, J., Zhang, Y.: NCRF++: An Open-source Neural Sequence Labeling Toolkit. In: Proc. of ACL 2018 (System Demonstrations). pp. 74–79 (2018)