Deep Learning Architecture for Patient Data De-identification in Clinical Records

Shweta, Asif Ekbal, Sriparna Saha, Pushpak Bhattacharyya
Indian Institute of Technology Patna
Bihar, India
{shweta.pcs14,asif,sriparna,pb}@iitp.ac.in

Abstract

Rapid growth in Electronic Medical Records (EMR) has emerged to an expansion of data in the clinical domain. The majority of the available health care information is sealed in the form of narrative documents which form the rich source of clinical information. Text mining of such clinical records has gained huge attention in various medical applications like treatment and decision making. However, medical records enclose patient Private Health Information (PHI) which can reveal the identities of the patients. In order to retain the privacy of patients, it is mandatory to remove all the PHI information prior to making it publicly available. The aim is to de-identify or encrypt the PHI from the patient medical records. In this paper, we propose an algorithm based on deep learning architecture to solve this problem. We perform de-identification of seven PHI terms from the clinical records. Experiments on benchmark datasets show that our proposed approach achieves encouraging performance, which is better than the baseline model developed with Conditional Random Field.

1 Introduction

With the phenomenal growth in medical interpretation, there have been tremendous increase of Electronic Medical Records (EMR) (Beck et al., 2012). Clinical documents contain valuable information (patient disease, medical procedure applied and medication) which have resulted in drawing good attention of researchers to explore and extract relevant information from the clinical text. However, these medical records consist of patient Private Health Information (PHI) (e.g., Patient name, Age, Doctor name, ID, Phone number, Address etc.) which can reveal the patient identity during the course of treatment. To avoid disclosing PHI information, it is mandatory according to the Health Insurance Portability and Accountability Act (HIPAA)[1], 1996, that the PHI terms are required to be hidden and protected prior to making it publicly available. De-identification is, thus, defined as the process of identifying and hiding PHI from clinical records and maintaining the integrity as much as possible (Stubbs et al., 2015). While during the course of PHI identification for removal, it is highly necessary for a de-identification process to retain the medical contents of the records so that this information can help further research and conserve the value of the record. However, de-identifying the records manually is quite unfeasible and expensive both in terms of time, efforts and cost. As such there is a huge requirement for an automated de-identification system.

De-identification task can be, in general, looked up as a traditional Named Entity Recognition (NER) task. Basically, NER can be thought of as a sequence labeling task with the goal to identify proper output sequences of the entities. Therefore, for every input sequence of words, the best labeled-sequence is to be obtained. De-identification task can be, in general, looked up as a traditional Named Entity Recognition (NER) task with the goal to identify proper output sequences of the entities. Therefore, for every input sequence of tokens, the best labeled-sequence is to be obtained. De-identification poses several challenges (Meystre et al., 2008). The two major hurdles for identifying PHI terms are as follows:

[1] http://www.hhs.gov/hipaa

Proceedings of the Clinical Natural Language Processing Workshop,
pages 32–41, Osaka, Japan, December 11-17 2016.

(1) Inter-PHI ambiguity: The ambiguity problem, where due to the lexical similarity, PHI terms overlap with the non-PHI terms. Example includes *Brown* (Doctor name) which is a PHI term vs. *brown* which is a non-PHI term.
(2) Intra-PHI ambiguity: This problem appears when one candidate word seems to belong to two different PHI terms. For example, the word *August* which is a Patient name vs. *August* which also denotes the possible candidate for date expression.
The problem of patient data de-identification has been addressed very recently in the shared task, Center of Informatics for Integrating Biology (i2b2) challenge[2]. The existing systems of patient data de-identification can be classified under three categories *viz.* rule-based, machine learning based and hybrid technique. Rule-based system follows patterns based on regular expression and gazetteers identified by the human. In practice, the set of rules corresponding to a system are restricted to a particular domain. Generally, the system fails when the domain is altered. To overcome this drawback, machine learning approaches have been proposed and found to be very successful in solving this de-identification problem. Some of the popular machine learning models proposed include support vector machine (Hara, 2006), (Guo et al., 2006), decision tree (Szarvas et al., 2006), log-linear models and most used conditional random fields (CRFs)(Yang and Garibaldi, 2015; He et al., 2015). Supervised machine learning and rule-based approach share the following drawbacks: these techniques require the labeled data and prominent feature set or the rules. This incurs cost and time as the appropriate set of features or rules can be framed only after analyzing the full records.
The advent of deep learning algorithms has facilitated to introduce a new framework where we do not require handcrafted features or rules. These models have the abilities to learn automatically the relevant features by performing composition over the words represented in the form of vectors known as word embedding. In recent times, deep neural network architecture has shown promise for solving various NLP tasks such as text classification (Socher et al., 2013; Kim, 2014), language modeling (Mikolov et al., 2010), question answering (Weston et al., 2015), machine translation (Bahdanau et al., 2014), spoken language understanding (Mesnil et al., 2013) etc. In this paper, we propose a novel system (DI-RNN) based on deep learning for patient data de-identification (PDI). We formulate the task as a sequence labeling problem and develop a technique based on Recurrent Neural Network (RNN) (Mikolov et al., 2010). RNN, unlike other techniques, does not require features to be explicitly generated for classifier's training or testing. Instead it learns features by itself which makes this approach domain adaptable and scalable. We develop a system for PDI in line with the framework introduced in Center of Informatics for Integrating Biology and the Bedside (i2b2) challenge[3]. The goal of the task was to identify all the PHI terms from the medical records. Firstly, we develop a baseline model based on a supervised machine learning algorithm, namely conditional random field (CRF) (Lafferty et al., 2001). The classifier is trained with a set of features automatically extracted from the training documents. We implement and compare different variants of RNN architectures, such as Elman-type networks (Elman, 1990; Mikolov et al., 2011) and Jordan-type networks (Jordan, 1997). The main aim of our paper is to study the effectiveness of deep learning techniques over the traditional supervised approaches for de-identification task.

2 Patient Data De-identification Task

The problem of patient data de-identification can be thought as a task equivalent to named entity recognition (NER). The main aim of both the tasks is to automatically identify noun phrases or part of noun phrases from the text. The problem of de-identification can be modeled as a two-step process, where in the first step all the PHI terms are required to be identified and classified, and in the later stage, identified PHI terms are encrypted. Here, we provide an example sentence with the corresponding NEs highlighted. Here, the input is the sequence of words W and the output corresponds to the sequence of labels L corresponding to the word-sequence and the corresponding de-identified sentence as shown in Table-1. Traditionally, the task can be visualized as follows: for a given word sequence W, the aim is to

[2]https://www.i2b2.org/
[3]https://www.i2b2.org/

Sentence	To	follow	up	with	Dr.	John	D	Doe
Named Entity	O	O	O	O	O	B-DOCTOR	I-DOCTOR	I-DOCTOR

Table 1: Examples of PHI instances represented by 'BIO' notation

find the best possible label-sequence that has maximum posterior probability i.e., $P(L|W)$. The Bayes rule is applied in the case of generative model framework as

$$\begin{aligned} \hat{L} &= argmax_L P(L|W) \\ &= argmax_L P(W|L)P(L) \end{aligned} \tag{1}$$

For the given sequence of words W, and its corresponding label sequence L, joint probability $P(W|L)P(L)$ has to be maximized by the objective function of a generative model.
Recently, Conditional Random Field (Lafferty et al., 2001), a discriminative model has become the popular technique for solving de-identification task (Yang and Garibaldi, 2015). Here, given the word sequence $W_1^N = w_1, ...w_N$, as input, CRF calculates the conditional probability of labels $L_1^N = l_1, ...l_N$, as follows:

$$P(l_1,l_2,...,l_N|W)=\frac{1}{Z_w}\prod_i(\Psi_i(L_i,W)\Psi_i^{'}(L_i,L_{i-1},W)) \tag{2}$$

where Ψ_i and $\Psi_i^{'}$ are defined as follows:

$$\Psi_i(L_i, W) = exp(\sum_k \eta_k s_k(l_i, w, i)) \tag{3}$$

$$\Psi_i^{'}(L_i, L_{i-1}, W) = exp(\sum_j \lambda_j t_j(l_i, l_{i-1}, w, i)) \tag{4}$$

where t_j and s_k are transition feature function and state feature function, respectively. The transition feature function t_j depends upon the current label l_i, previous label l_{i-1} and the observation sequence of word w at time i. The state feature function is the function of current label l_i and the observation word w at time i. Parameters λ_j and η_k are to be estimated from training data.
Other variants of discriminative models include Support Vector Machines (SVMs) (Cortes and Vapnik, 1995), where local probability functions are used. With these traditional methodologies, classification algorithm is a black box implementation of linear and log-linear approaches which require good feature engineering. After conducting thorough literature survey, deep learning architecture is found to be one of the successful techniques where both classification and feature designing are done during the learning phase automatically without using any human intervention. Therefore, we propose a technique based on deep learning architecture of RNN. We discuss below the RNN architecture with respect to our chosen problems.

3 Proposed Approach for Patient Data De-identification

The RNN models used for de-identification task are described here.

3.1 Word Embedding

A real-valued representation of a word is the input for our RNN architecture. Word embedding provides an unique property to capture semantics and syntactic information of different words (Mikolov et al., 2013). The underlying idea is that similar words appear in close vicinity of each other. The vector corresponding to each input word w_i is produced whose dimensionality is set at the time of learning the neural language model from the given unsupervised corpus. This representation provides the continuous-space representation for each word. Usually, training of the word embedding is done in an unsupervised manner using external natural language text like Wikipedia, news article, bio-medical literature etc. The architecture can be varied by adopting various architectures like shallow neural networks (Schwenk and Gauvain, 2005), RNN (Mikolov et al., 2010; Mikolov et al., 2011), SENNA (Collobert et al., 2011),

word2vec (Mikolov et al., 2013) etc. We use three different procedures to learn word embeddings like random number initialization, RNN's word embedding and continuous bag-of-words (CBOW). For random word embedding we initialize the vector of dimension 100 in the range -0.25 to $+0.25$. In order to evaluate the impact of RNN we use the word embedding as provided by RNNLM [4] of dimension 80 which is trained on Broadcast news corpus. In addition to this we also use word embedding model trained by CBOW technique as proposed in (Mikolov et al., 2013) on news data of dimension 300.

3.2 Word Dependencies captured using a Context Window

In feed forward neural network model we provide input as word embedding of the target word. But, it can not capture the dependency associated with the current word of interest. Context words can capture the short-term temporal dependencies in this setting. Let us assume that each word is being represented by its word embedding vector of length d, the word-context window is the ordered concatenation of word embedding vectors. For word embedding of dimension d and context word of size m, the word vector is constructed as the ordered concatenation of $2m + 1$ word embedding vectors, i.e. m previous words, current word and m next words with the following formula

$$C_m(x_{i-m}^{i+m}) = v_{i-m} \oplus \ldots v_i \ldots \oplus v_{i+m} \tag{5}$$

where $\oplus$ is a concatenation operator. v_i is the word embedding vector of the word x_i.
$x_{i-m}^{i+m} = [x_{i-m} \ldots, x_i, \ldots x_{i+m}]$ represents the concatenation of dependent words in the window size m. In order to generate m context window for the beginning and ending words, padding is performed. We provide an example below to show the generation of context window of size 1 around the word 'suffers':

$$C(t) = [\text{Doe } \textit{suffers} \text{ from}] \tag{6}$$

$$C(t) \rightarrow x(t) = [v_{\text{Doe}}\ v_{\textit{suffers}}\ v_{\text{from}}]$$

In this example, $C(t)$ is a 1 word context window. $v_{\textit{suffers}}$ is the embedding vector of word 'suffers' and d is the dimension of the embedding vector. Similarly, $C(t)$ forms the ordered concatenation of word embedding vector for the word sequence $x(t)$ at time t.

3.3 Variants of RNN Architecture: Elman and Jordan

In this section, we discuss two different variants of RNN architecture, Elman (Elman, 1990) and the Jordan models (Jordan, 1997). Figure-1 depicts an architecture for both the models. Feed forward neural network (NN) (Svozil et al., 1997) is the basic biologically inspired neural network model. In variation to feed forward architecture, both the RNN models make connection also with the previous layer. In Elman architecture each state keeps track of its previous hidden layer states by its recurrent connections. Therefore, the hidden layer $h(t)$ at time instance t keeps track of the previous $(t-1)^{th}$ hidden layer i.e., the output of $(t-1)^{th}$ hidden layer is given as the input to the t^{th} hidden layer $h(t)$ along with the context window input $C_m(x_{t-m}^{t+m})$. Mathematically, for H hidden layer, Elman architecture is described as shown below:

$$h^{(1)}(t) = f(W^{(1)} C_m(x_{t-m}^{t+m}) + V^{(1)} h^{(1)}(t-1) + b) \tag{7}$$

$$h^{(H)}(t) = f(W^{(H)} h^{(H-1)}(t) + V^{(H)} h^{(H)}(t-1) + b) \tag{8}$$

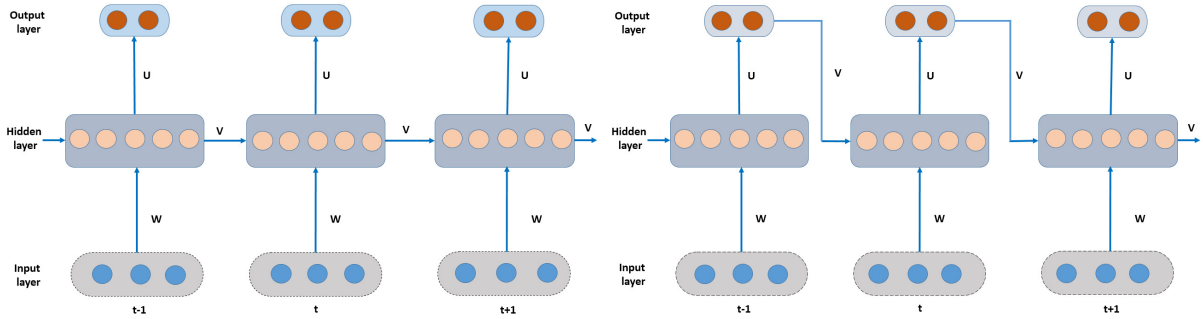
In our experiment we have used a non-linear sigmoid function as the activation unit of hidden layer.

$$f(x) = 1/(1 + e^{-x}) \tag{9}$$

The superscript represents the hidden layer depth and, W and V denote the weight connections from input layer to the hidden layer and hidden layer of last state to current hidden layer, respectively. Here, b is a bias term. The softmax function is later applied to the hidden states to generate the posterior probabilities of the classifier for different classes as given below:

$$P(y(t) = i | C_m(x_{t-m}^{t+m})) = g(U h^{(H)}(t) + c) \tag{10}$$

[4] http://rnnlm.org/



(a) Elman Architecture (b) Jordan Architecture

Figure 1: RNN architectures of both the variants

Here, U is weight connection from hidden to output layer, c is a bias term and g is softmax function defined as follows:

$$g(z_m) = \frac{e^{z_m}}{\sum_{i=1}^{i=k} e^{z_k}} \tag{11}$$

Jordan model is the another variation of RNN architecture which is similar to the Elman model except the input to the recurrent connections are through the output posterior probabilities:

$$h(t) = f(WC_m(x_{t-m}^{t+m}) + VP(y(t-1)) + b) \tag{12}$$

where W and V denote the weight connection between input to hidden layer and output layer of previous state to current hidden layer and $P(y(t-1))$ is the posterior probability of last word of interest. The sigmoid function described in Eq-9 is used as non-linear activation function f.

3.4 Datasets

The dataset used to evaluate our proposed architecture is obtained from 2014 I2b2 challenge (Stubbs et al., 2015). This dataset is obtained from "Research Patient Data Repository of Partners Healthcare". A total of 1304 medical records were manually annotated. In order to use this data for our experiment we split the data set into three parts: training, validation and test. The detailed distribution of different PHI terms in these three sets are described in Table-2.

Our training data compromises of 11,911 PHI relevant instances, while the test dataset consists of total 1253 PHI instances which we developed from I2B2-2014 training data. To ensure the patient confidentiality as much as possible, the challenge aims to identify HIPAA-PHI categories firstly with the added subcategories. This dataset is annotated using seven main PHI categories with the twenty-five associated subcategories. While, our experiments cover the seven main PHI categories, I2b2 challenge covers almost all HIPAA defined categories and subcategories. The list of categories as well as subcategories are 1. Name (subtypes: Patient, Doctor, Username), 2. Profession, 3. Location (subtypes: Hospital, Department, Organization, Room, Street, City, State, Country, ZIP), 4. Age, 5. Date, 6. Contact (subtypes: Phone, Fax, Email, URL, IPAddress), 7. Ids (subtypes: Medical Record Number, Health Plan Number, Social Security Number, Account Number, Vehicle ID, Device ID, Licence Number, Biometric ID). In this work, the aim is to identify seven different PHI subtypes; *Patient, Doctor, Hospital, Location, Phone, ID* and *Date* from the above defined categories. In order to evaluate the model performance well known evaluation metrics such as recall, precision and F-Measure are used.

3.5 RNN Hyper-Parameters and Learning

The RNN hyper-parameters are number of hidden units (H), learning rate (λ), context window size (m), no. of epochs (e^n) and dropout probability (p). In order to find optimal hyper-parameter values we experiment with different parameter settings. The optimal hyper-parameter values for both the RNN architectures are listed in Table-3. The embedding matrix and the weight matrices are initialized from

PHI category	Train	Validation	Test
DOCTOR	2262	183	236
HOSPITAL	1342	141	164
DATE	4154	377	498
PATIENT	707	28	59
LOCATION	93	14	19
PHONE	153	12	13
ID	3200	233	264
Total	11911	988	1253

Table 2: Data set statistics: distribution of different classes for training, test and validation sets.

Parameter's	E-RNN	J-RNN
Hidden layer size	100	150
learning rate	0.01	0.01
Dropout probability	0.5	0.5
no. of epochs	25	25
context window size	11	9

Table 3: Optimal values of hyper-parameters for different RNN architectures

the uniform distribution in the range [-1,1]. In order to train RNN we use stochastic gradient descent. We consider the whole sentence as a mini-batch and perform one update per sentence, towards minimizing the negative log-likelihood.

3.6 Regularization

In order to prevent network from over-fitting we use dropout technique (Hinton et al., 2012). Dropout omits the portion of hidden unit from each training sample before passing it to the final softmax layer. We set dropout probability p as 0.5 throughout the experiments in both the variations of RNN.

3.7 Impact of Word Embedding Techniques

Table-4 shows the impact of each word embedding techniques with Elman architecture. The word vectors obtained from the RNNLM performed well on syntactic part. It is obvious because the word vectors in the RNNLM are directly connected to a non-linear hidden layer. The CBOW architecture works better than RNNLM for the syntactic tasks, and about the same on the semantic tasks. The CBOW model follows the distributional hypothesis while training which enables to outperform over the other word embedding techniques.

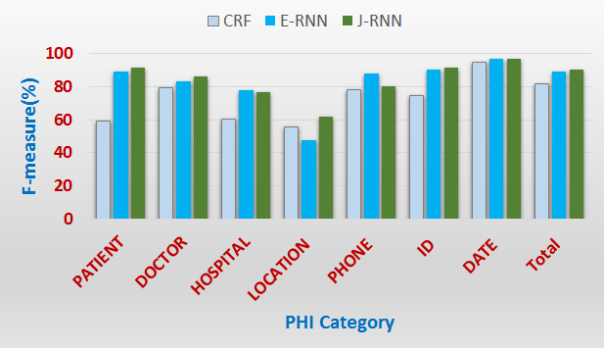


Figure 2: Performance comparisons between RNN and CRF for all identified PHI types

Word Embedding Techniques	dimension (d)	Precision	Recall	F-Measure
Random Number	100	87.19	85.48	86.32
RNNLM	80	88.21	87.32	87.76
CBOW	300	89.35	89.55	89.44

Table 4: Impact of fine-tuned word embedding technique using Elman architecture. Here RNNLM: Word embedding obtained from the RNN language modeling technique(Mikolov et al., 2010). CBOW: The CBoW takes the context word as input and tries to predict the target word.

3.8 Results with Lexical Features

In the literature there are quite a few works of patient data de-identification using lexical features such as PoS, character n-gram, chunk information etc. In the literature, it has been shown that CRF is a robust classifier for this task. In addition to the RNN we also perform experiments with some useful hand-crafted features by considering CRF as the base classifier. The hand-crafted features that we use for CRF are as follows:

1. Context word feature: We use current word and the words within the context window of length 3 as features.

PHI category	CRF Baseline			E-RNN			J-RNN		
	R	P	F	R	P	F	R	P	F
PATIENT NAME	60.87	57.14	58.95	86.96	90.91	88.89	91.30	91.30	91.30
DOCTOR NAME	80.43	77.78	79.08	82.55	83.98	83.26	85.11	86.58	85.84
HOSPITAL NAME	47.24	83.70	60.39	73.01	83.80	78.03	70.55	83.33	76.41
LOCATION	52.63	58.82	55.56	57.89	40.74	47.83	68.42	56.52	61.90
PHONE	69.23	90.00	78.26	84.62	91.67	88.00	76.92	83.33	80.00
ID	75.86	73.06	74.44	89.27	91.37	90.31	90.80	92.58	91.68
DATE	95.17	94.22	94.69	98.39	95.14	96.74	98.39	95.32	96.83
Overall	**79.74**	**83.11**	**81.39**	**88.90**	**89.55**	**89.22**	**89.63**	**90.73**	**90.18**

Table 5: Detailed performance analysis with different models for PHI identification task. Here **R**,**P** and **F** denotes *Recall, Precision* and *F-score* respectively.

2. Bag-of-word feature: This feature includes uni-grams, bi-grams, tri-grams of the target token. We use window size of [-2, 2] with respect to the target token. Here, n-gram is referred as the continuous sequence of n items. An n-gram generated having sizes of $1, 2, 3$ are known as an uni-gram, bi-gram and tri-gram, respectively.
3. Part-of-Speech (PoS) Information: The PoS information of current word, previous two words and the next two words are used as features. We obtain PoS of words from the Stanford tagger (Toutanova and Manning, 2000).
4. Chunk Information: The chunk information is an important feature to identify the PHI term-boundary. We use chunk information obtained from *openNLP*[5].
5. Combined POS-token and Chunk-token Feature: This feature is generated by the combination of other token features like PoS, Chunk within the context window of [-1, 1]. This is represented as $[w_0p_{-1}, w_0p_0, w_0p_1]$ where w_0 represents the target word, and p_{-1}, p_0 and p_1 represent the previous, current and the next POS or Chunk tags, respectively.
We build our model by incorporating the above features. We use the CRF implementation[6] of CRF^{++} with default parameter settings. Detailed results on PHI identification task using these features with CRF classifier are shown in Table-5.

3.9 Results with RNN

The Elman architecture that we discussed in Subsection-3.3 has been applied to identify the PHI terms from medical records. Table-5 shows the detailed results of E-RNN on individual PHI categories as well as the overall results. The E-RNN performs better than our CRF baseline model. The experiments are performed with all the types of word embedding techniques discussed in Subsection-3.7. The CBoW based word embedding, when given as input to E-RNN model, performs well over the other word embedding based techniques as shown in Table 4. Experimental results on Jordan architecture are shown in Table-5. The performance that we obtain shows better performance over the baseline. We show detailed comparative results in Table-5. Experiments reveal that J-RNN model performs superior compared E-RNN in identifying 5 PHI categories out of total 7.

4 Error Analysis

We perform detailed error analysis on outputs produced in both the models. We divide the major sources of errors in three different categories. Following observations can be made:

[5]https://opennlp.apache.org/
[6]https://taku910.github.io/crfpp/

- MISSED ENTITY: This error occurs when the entity is present in the gold-standard data, but the system fails to identify it as an entity. We calculate a total of 106 and 95 cases in Elman and Jordan model, respectively, for such cases. The possible causes are:
  - Presence of single-word person name: These words are difficult to detect as compared with full names (consist of more than one words) due to the lack of context and morphology. These errors are more dominated in case of 'Doctor' and 'Patient' categories.
  - Presence of abbreviated words: These errors are dominated mostly in case of 'Hospital' and 'Doctor' categories as the system lacks in identifying the short words (e.g., "FIH", "WA") due to the presence of ambiguous non-PHI terms.
  - Presence of unseen terms: The words not seen during training contribute to this error. These cases are mostly found for 'Location', and 'Hospital' categories.

- WRONG ENTITY: This error is obtained when the entity obtained is correct but belongs to some other type. In total 223 and 164 instances are mis-classified in case of Elman and Jordan model, respectively. The major causes of actual errors are as follows:
  - Inter-PHI ambiguity: These errors are obtained mostly in case of 'Doctor' and 'Patient' categories. As the name-forms are quite similar to each other, these PHI terms are highly ambiguous. This error arises most of the times when the names consist of single words. For example "Glass", "Chabechird" etc. These cases are also observed in case of 'Location' category.

- FALSE POSITIVES: This error occurs when the system lacks in identifying the proper boundary of the entity. Either the entity has additional part or the missing part. These errors are mostly seen in case of 'Doctor' and 'Hospital' categories. The major cause of this error is:
  - Presence of long compounded words: If the entity consists of more than 3 words, the system fails to identify those correctly. For example "Tawn List Medical Center".

4.1 Discussions

Two different RNN architectures, E-RNN and J-RNN, perform well over the baseline model based on machine learning technique. The J-RNN outperforms the E-RNN model in most of the PHI category detection. The J-RNN model takes the outputs of previous iteration along with the outputs of current hidden layer to classify the current word. It would be the possible reason behind the better system performance for strict[7] PHI (Patient, Doctor) as compared to the performance of E-RNN for the same. It should be noted that due to computational limitation, we were not able to use whole dataset as such we were unable to make any direct comparison with the existing systems. Most of the existing systems are supervised in nature and makes use of hand-crafted feature set and rules. These techniques require much feature engineering. The development of quality features are challenging and time-consuming. In our case, we don't use any hand-crafted feature set, but still achieves good performance level.

5 Conclusions and Future Works

In this paper we present a deep neural network based approach for patient data de-identification. This has been designed to identify and classify Protected Health Information (PHI) present in free-text medical records and encrypt these for preserving the privacy of patients. We systematically implement and compare different variants of RNN architecture, including Elman and Jordan. In order to compare we also develop a CRF based model with the traditional features. We observe that both the variants of RNN architecture outperform the baseline built using popular CRF based model. We have observed the performance improvement of 7.83% with Elman and 8.79% with Jordan over the baseline model. In future, we would like to explore more advanced deep learning techniques like Long Short term Memory (LSTM) using the full dataset and on other domains as well.

[7]Since it is a kind of multiword NE's, in which previous label information is vital to identify the current

Acknowledgements

Authors gratefully acknowledge for the partial support received from "Sushrut: ezDI Research Lab on Health Informatics", Department of Computer Science and Engineering, IIT Patna, India.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Tim Beck, Sirisha Gollapudi, Søren Brunak, Norbert Graf, Heinz U Lemke, Debasis Dash, Iain Buchan, Carlos Díaz, Ferran Sanz, and Anthony J Brookes. 2012. Knowledge engineering for health: a new discipline required to bridge the ict gap between research and healthcare. *Human mutation*, 33(5):797–802.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Yikun Guo, Robert Gaizauskas, Ian Roberts, George Demetriou, and Mark Hepple. 2006. Identifying personal health information using support vector machines. In *i2b2 workshop on challenges in natural language processing for clinical data*, pages 10–11. Citeseer.

Kazuo Hara. 2006. Applying a svm based chunker and a text classifier to the deid challenge. In *i2b2 Workshop on challenges in natural language processing for clinical data*, pages 10–11. Am Med Inform Assoc.

Bin He, Yi Guan, Jianyi Cheng, Keting Cen, and Wenlan Hua. 2015. Crfs based de-identification of medical records. *Journal of biomedical informatics*, 58:S39–S46.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Michael I Jordan. 1997. Serial order: A parallel distributed processing approach. *Advances in psychology*, 121:471–495.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289.

Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *INTERSPEECH*, pages 3771–3775.

Stéphane M Meystre, Guergana K Savova, Karin C Kipper-Schuler, John F Hurdle, et al. 2008. Extracting information from textual documents in the electronic health record: a review of recent research. *Yearb Med Inform*, 35:128–44.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3.

Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5528–5531. IEEE.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Holger Schwenk and Jean-Luc Gauvain. 2005. Training neural network language models on very large corpora. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 201–208. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer.

Amber Stubbs, Christopher Kotfila, and Özlem Uzuner. 2015. Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/uthealth shared task track 1. *Journal of biomedical informatics*, 58:S11–S19.

Daniel Svozil, Vladimir Kvasnicka, and Jiri Pospichal. 1997. Introduction to multi-layer feed-forward neural networks. *Chemometrics and intelligent laboratory systems*, 39(1):43–62.

György Szarvas, Richárd Farkas, and András Kocsor. 2006. A multilingual named entity recognition system using boosting and c4. 5 decision tree learning algorithms. In *International Conference on Discovery Science*, pages 267–278. Springer.

Kristina Toutanova and Christopher D Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pages 63–70. Association for Computational Linguistics.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Hui Yang and Jonathan M Garibaldi. 2015. Automatic detection of protected health information from clinic narratives. *Journal of biomedical informatics*, 58:S30–S38.