A Recurrent Neural Network Architecture for De-identifying Clinical Records

Shweta, Ankit Kumar, Asif Ekbal, Sriparna Saha, Pushpak Bhattacharyya
Indian Institute of Technology Patna
Bihar, India
{shweta.pcs14,ankit.cs13,asif,sriparna,pb}@iitp.ac.in

Abstract

Electronic Medical Records contains a rich source of information for medical finding. However, the access to the medical record is limited to only de-identified form so as to protect the confidentiality of patient. According to Health Insurance Portability and Accountability Act, there are 18 PHI categories that should be enclosed before making the EMR publicly available. With the rapid growth of EMR and a limited amount of de-identified text, the manual curation is quite unfeasible and time-consuming, which has drawn the attention of several researchers to propose automated de-identification system. In this paper, we proposed deep neural network based architecture for de-identification of 7 PHI categories with 25 associated subcategories. We used standard benchmark dataset from i2b2-2014 de-identification challenge and performed the comparison with very strong baseline based on Conditional Random Field. We also perform the comparison with the state-of-art. Results show that our proposed system achieves significant improvement over baseline and comparable performance over state-of-art.

1 Introduction

Appreciable amount of information extracted from Electronic Medical Record (EMR) have flourished Medical Natural Language Processing in recent past. In general, the medical records are restricted according to Health Insurance Portability and Accountability Act (HIPAA)[1], 1996. Before making it publicly available, the medical records should be de-identified which refers to hiding the personal details. De-identification can be thus seen as the task of enclosing the private health information (PHI) while maintaining the exact sense of the record. According to the HIPAA standards, total of 18 PHI categories have to enclosed before making records publicly available. Taking into account, the vast size of available EMR, manual de-identification could be expensive and unfeasible. These motivate us to develop an automated de-identification system for this task.

De-identification shares the common property with the traditional named entity recognition which aims to identify the proper labeled sequence for the given input sequence. However, detection of PHI entities suffers from several challenges such as:

(1) Terminological variation and irregularities: PHI entities can occur within text in different variations, for example '3041023MARY' is the combination of two different PHI categories '3041023' which represents the MEDICALRECORD and 'MARY' which is another PHI category.

(2) Lexical variations: In EMR same entities are often written in different lexical form. For example, variation of the entities such as the '50 yo m', '50 yo M', '55 YO MALE'.

(3) Inter-PHI ambiguity: Ambiguity of PHI terms with the non-PHI terms. For e.g., 'Brown' can be identified as the PHI term 'Name (Doctor)' as well as non-PHI term.

(4) Intra-PHI ambiguity: Ambiguity of PHI terms with the other PHI terms. For e.g., '30s' can be identified as the PHI term (Age) as well as other PHI terms (Date).

Recently several shared tasks have been organized to solve the de-identification problem such as Center of Informatics for Integrating Biology (i2b2)[2].

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

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

D S Sharma, R Sangal and A K Singh. Proc. of the 13th Intl. Conference on Natural Language Processing, pages 188–197,
Varanasi, India. December 2016.

The traditional de-identification system generally falls into three different categories *viz.* machine-learning-based system, rule based system and hybrid system (based on the machine learning and rule based). Rule based system depends on the patterns formed by the regular expressions and gazetteers which are developed by humans. Rule based techniques might be very successful for one domain but fail to show significant improvements when domain changes. To overcome these difficulties, supervised machine learning techniques were proposed to solve the de-identification task. The popular machine learning models were based on decision tree (Szarvas et al., 2006), support vector machine (Hara, 2006), (Guo et al., 2006), log-linear models and popular conditional random fields (Yang and Garibaldi, 2015; He et al., 2015). However, existing techniques based on machine learning suffer from the following drawbacks: (1) requirement of significant amount of labeled data, (2) involves an extensive feature engineering or rule generation step necessitating human effort. Hence, both the techniques require manual intervention for designing features and rules which are restricted to single domain and thus incur time and cost.

The introduction of deep learning technique has facilitated to learn effective features without any manual intervention i.e., there is no requirement of feature engineering. The models could learn implicitly relevant features by word in the form of vectors known as the word embedding. These embedding are jointly learned by other hyper-parameters which are initialized randomly or can be pre-trained on large unlabeled corpus. Pre-training is much beneficial in improving performance as it effectively captures the linguistic variations and patterns. Recently, there has been significant success of deep learning techniques in solving various natural language processing tasks such as text classification (Kim, 2014), language modeling (Mikolov et al., 2010), machine translation (Bahdanau et al., 2014), spoken language understanding (Mesnil et al., 2013) as well as named entity recognition (Collobert et al., 2011; Lample et al., 2016).

Motivated by the success of deep learning techniques, in this paper, we have adopted in particular Recurrent Neural Network (RNN) (Mikolov et al., 2010) architecture to capture PHI terms. RNN has shown advantages over other machine learning and rule based techniques. RNN unlike other techniques does not require features explicitly developed for the classifier learning. The virtue of system learning by itself makes the system adaptable and scalable. This work is an extension of our previous work (Shweta et al., 2016) where we identified only 7 PHI category (Patient, Doctor, Hospital, Location, Date, Age, ID) irrespective of subcategories using only i2b2-2014 training dataset. The current work provide comprehensive experimentation on i2b2-2014 challenge dataset to de-identify 7 categories and 25 subcategories. We have formulated this task as the sequence labeling problem and developed the baseline model using a supervised machine learning technique. Conditional random field (CRF) (Lafferty et al., 2001) along with a set of handcrafted features are used to build the base classifier.

In the current study, we performed comparative analysis with two different variants of RNN network model *viz* Elman-type networks (Elman, 1990; Mikolov et al., 2011) and Jordan-type networks (Jordan, 1997). A thorough comparison of these two RNN variants with strong baseline based on CRF is a part of the paper. The results obtained show the effectiveness of RNN over traditional CRF based model. We further compared our deep learning model with state-of-art results on de-identification task. We have shown that RNN achieves comparable results with the state-of-art using machine learning techniques.

2 Related Works

Since last decade, de-identification task has emerged as a fascinated research problem (Coorevits et al., 2013). Recently, various challenges have been organized for this task. Center of Informatics for Integrating Biology and the Bedside (i2b2) has organized several de-identification shared tasks. In i2b2 2006 shared task (Uzuner et al., 2007), Wellner et al.(2007) achieved the remarkable performance by adapting machine learning approach using CRF and SVM as the base classifiers with some lexical and semantic features. Szarvas et al.(2007) developed an iterative technique using machine learning based approach. They designed local features and used dictionaries for learning decision tree based classifier. Most of the submitted systems used Conditional Random Field (CRF) classifier (Wellner et al., 2007; Aramaki et al., 2006), while some systems had also

used SVM (Hara, 2006). Most of the submissions focused on the machine learning techniques while some systems (Guillen, 2006) made use of rule based approaches for solving this task.

In 2014 I2b2 shared task, the task was relatively stricter than 2006 shared task. Here the challenge was to identify 8 PHI categories with the associated subcategories. Yang et al.(2015) developed best performing system. They adopted hybrid technique considering both machine learning and rule based techniques. They developed several features like linguistic, syntactic and various word surface oriented features with different regular expressions to capture PHI terms like date and ID. Dehghan et al.(2013) developed system using knowledge based and rule based approaches using CRF as classifier. Xu et al.(2010) utilized the biomedical dictionary for identifying the PHI terms. Literature survey shows that hybrid systems perform better over the rule based and machine learning based techniques.

3 De-identification of Electronic Medical Record

De-identification of EMR can be identified as a two phase task, where the first phase of the task deals with the extraction and classification of entities (PHI) from the medical records and second phase deals with the encryption of identified PHI terms. In the current study the first phase of the problem is formulated as a sequence labeling task while some of the existing systems treat this as a classification problem.

We visualize this task as the traditional named entity recognition task, where for the given word sequence W, the goal is to identify the best possible label sequence L with the maximum posterior probability represented as $P(L|W)$. In case of generative model framework, Bayes rule can be applied as

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

Thus for each W and L, joint probability $P(W|L)P(L)$ has to be maximized by the objective function of a generative model. Table-1 shows the input as word-sequence with its corresponding label sequence and the output as the de-identified sentence.

Several probabilistic models, like SVM, HMM and most popular CRF model, have been used for solving sequence labeling problem in the literature.

In this work, we have developed CRF based model as the baseline. Here, each patient note is first pre-processed which includes tokenization and feature generation for each corresponding token. During training, CRF parameter is optimized to maximize the posterior probability while during test phase, the best output label is predicted. Several systems based on CRF were introduced in i2b2-2014 challenge which performed well in de-identification task. Other discriminative models such as Support Vector Machines (SVM) (Cortes and Vapnik, 1995) are very popular where local probability functions are used. Other popular models include Hidden Markov Model (HMM) (Rabiner and Juang, 1986). However, these models require a good feature engineering which is mostly applicable for a single domain. This motivated us to use Recurrent Neural Network architecture for solving the patient de-identification task.

4 RNN Architecture for De-identification

We describe here recurrent neural network (RNN) architecture w.r.t de-identification of EMR.

4.1 Neural network based Word Representation: Word Embedding

Word embedding is real valued word representation in the form of a vector. This vector is provided as input to the RNN architecture. Word embedding thus have powerful capability to capture both semantic and syntactic variations of words (Mikolov et al., 2013). The vector initially can be generated randomly or can be pre-trained from the large unlabeled corpus in an unsupervised fashion using external resources such as Wikipedia, news article, bio-medical literature etc. Word embedding is learned through sampling word co-occurrence distribution. These techniques are useful to identify similar words which appear in close vicinity in vector space. There are several ways of generating the word-vectors using different architectures such as word2vec (Mikolov et al., 2013), shallow neural networks (Schwenk and Gauvain, 2005), RNN (Mikolov et al., 2010; Mikolov et al., 2011) etc. We learn our word embedding through three different ways such as random number initialization, RNN's word embedding and continuous bag-of-words (CBOW) based models. In case

Sentence	Discussed	this	case	with	Dr.	John	Doe	for	Mr.	Ness
Named Entity	O	O	O	O	O	B-DOCTOR	I-DOCTOR	O	O	B-PATIENT
De-identified Sentence	Discussed	this	case	with	Dr.	XYZ_DOCTOR		for	Mr.	XYZ_PATIENT

Table 1: Sample sentence (sequence of words with the corresponding labels using BIO notation) and its corresponding de-identified sentence

of random number initialization, we randomly generate vector of length 100 in the range -0.25 to $+0.25$ for each word. To exploit the significance of RNN, we have used the word embedding of dimension 80 for a word trained on Broadcast news corpus as provided by RNNLM [3]. In addition to these we have also generated 300 dimension vector for a word trained using CBOW technique (Mikolov et al., 2013) on news corpus.

4.2 Capturing Short term Dependency with Context Window

The input provided to feed forward neural network is the word embedding of a target word. However, just the target word lacks in effectively capturing the dependencies related to the target word. While, context words are very helpful in capturing short-term temporal dependencies. As such for each word, d dimensional word embedding is generated with the word-context window of size m. We generate the word vector as the ordered concatenation of $2m+1$ word embedding vectors considering m previous words, m next words and current word as follows:

$$C_m(w_{i-m}^{i+m}) = v_{i-m}^d \oplus \dots v_i^d \dots \oplus v_{i+m}^d \quad (2)$$

Here, $\oplus$ is a concatenation operator where for each word w_i, the word embedding vector v_i is generated. Within the window size m, concatenation of dependent words is represented as follows: $w_{i-m}^{i+m} = [w_{i-m} \dots, w_i \dots w_{i+m}]$ For the words in the beginning and end, padding is performed in order to generate m context window. Below shows an example for context window 2 generation for the target word 'Hess'

$$C(t) = [\text{for Clarence } \textit{Hess} \text{ at BCH}] \quad (3)$$

$$C(t) \rightarrow x(t) = [v_{\text{for}}^d \; v_{\text{Clarence}}^d \; v_{\textit{Hess}}^d \; v_{\text{at}}^d \; v_{\text{BCH}}^d]$$

Here, $C(t)$ represents context window of 2 words. $v_{\textit{Hess}}$ denotes the word embedding vector for the target word 'Hess' and the embedding vector dimension is provided by d. Similarly, for each sequence of word $w(t)$ at t time, their vector concatenation is represented by $C(t)$.

[3] http://rnnlm.org/

4.3 Variant of RNN Model

Here we have used two different variants of RNN architecture for de-identification of patient notes. These are Elman-type RNN (Elman, 1990) and the Jordan-type RNN (Jordan, 1997). Architecture for both the models have been depicted in Figure-1. The neural network architecture is motivated from the biological neural network. The basic neural network is the feed forward neural network (NN) (Svozil et al., 1997) model. In contrast to the basic feed forward model, the connection formed in RNN is also through the previous layers. In Elman-type network, every state have the information of its previous hidden layer states through its recurrent connections. As such, the hidden layer $h(t)$ at the time instance t have the information of the previous $(t-1)^{th}$ hidden layer i.e., the output of $(t)^{th}$ hidden layer is dependent on the $(t-1)^{th}$ hidden layer $h(t-1)$ and context window $C_m(w_{t-m}^{t+m})$ as input. Below provide the mathematical expression for Elman-type network with H hidden layers

$$h^{(1)}(t) = f(\mathbf{W}^{(1)} C_m(w_{t-m}^{t+m}) + \mathbf{U}^{(1)} h^{(1)}(t-1) + \mathbf{b}) \quad (4)$$

$$h^{(H)}(t) = f(\mathbf{W}^{(H)} h^{(H-1)}(t) + \mathbf{U}^{(H)} h^{(H)}(t-1) + \mathbf{b}) \quad (5)$$

A non-linear sigmoid function as the activation unit of hidden layer has been used throughout the experiments.

$$f(x) = 1/(1+e^{-x}) \quad (6)$$

The superscript represents the hidden layer depth and, $\mathbf{W}$ and $\mathbf{U}$ denote the weight connections from input layer to the hidden layer and hidden layer of last state to current hidden layer, respectively. Here, $\mathbf{b}$ 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:

$$P(y(t) = i | C_m(w_{t-m}^{t+m})) = g(\mathbf{V} h^{(H)}(t) + \mathbf{c}) \quad (7)$$

Here, $\mathbf{V}$ is weight connection from hidden to output layer, $\mathbf{c}$ is a bias term and g is the softmax

function defined as follows:

$$g(z_m) = \frac{e^{z_m}}{\sum_{i=1}^{i=k} e^{z_k}} \quad (8)$$

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

$$h(t) = f(\mathbf{W}C_m(w_{t-m}^{t+m}) + \mathbf{U}P(y(t-1)) + \mathbf{b}) \quad (9)$$

where $\mathbf{W}$ and $\mathbf{U}$ denote the weight connection between input to hidden layer and output layer of previous state to current hidden layer, respectively, and $P(y(t-1))$ is the posterior probability of last word of interest. The sigmoid function described in Eq-6 is used as non-linear activation function f.

5 Dataset, Experiments and Results

In the current study, we have used the standard benchmark dataset of i2b2-2014 challenge (Stubbs et al., 2015) to evaluate our model. The challenge was part of 2014 i2b2/UTHealth shared task Track 1 (Stubbs et al., 2015). Total ten teams have participated in the shared task resulting in 25 different submissions. The i2b2-2014 dataset is the largest publicly available de-identification dataset collected from "Research Patient Data Repository of Partners Healthcare". A total of 1304 medical records of 297 patients were manually annotated which were divided into training and test set comprising of 790 and 514 records, respectively. There are $17,045$ and $11,462$ PHI instances in the training and test sets, respectively. This was manually annotated using seven types with twenty-five subcategories as: **(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, License Number, Biometric ID) Table-2 provides detailed distribution of PHI terms in both the sets.

PHI category	Train	Test
NAME	2262	2883
PROFESSION	234	179
LOCATION	2767	1813
AGE	1233	764
DATE	7502	4980
CONTACT	323	218
ID	881	625

Table 2: Data set statistics: distribution of different classes in training and test sets.

5.1 Evaluation measures

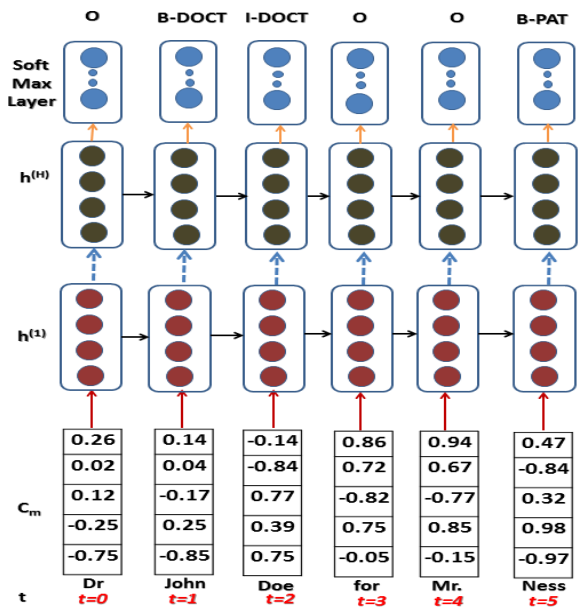
For the evaluation of our model, we adopted similar evaluation metrics as used in i2b2 challenge such as recall (R), precision (P) and F-Measure (F). *recall* is defined as the ratio of total number of correctly predicted PHI terms by model with the total PHI terms available in gold data. Similarly *precision* is the ratio of correctly predicted PHI terms by model with the total number of PHI terms predicted my model. The *F-measure* is the harmonic mean of precision & recall. We have computed these values at the entity level across the full corpus. Micro-averaged F-measure is used as our primary metric. This helps in identifying how system performs compared to gold standard data. We have used the same i2b2 evaluation script to make comparative analysis with the existing systems.

5.2 Learning Methods: Fine tuning RNN hyper-parameter

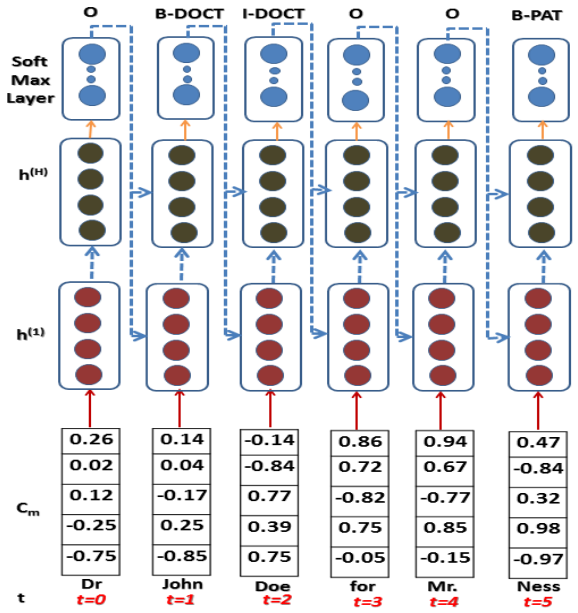
We have trained our RNN model using stochastic gradient descent. RNN can be tuned with hyper-parameters such as number of hidden layers (H), context window size (m), learning rate (λ), dropout probability (p) and no of epochs. In order to fine tune our system, we have conducted experiments on development set which is 10 % of our training data. For training the RNN model, we have performed mini batch gradient descent approach considering only one sentence per mini batch, minimizing negative log-likelihood. We have initialized the embedding and weight matrices in the range of $[-1, 1]$ following uniform distribution. Table-3 shows the optimized hyper-parameter values for both the RNN models.

5.3 Dropout Regularization

Over-fitting causes the degradation of system performance in RNN model. In order to prevent this, we have used recently proposed regularization technique know as dropout (Hinton et al., 2012). Dropout excludes some portion of hidden layers as well as the input vector from every



(a) Elman Architecture



(b) Jordan Architecture

Figure 1: Architecture of Recurrent Neural Network: Elman & Jordan type. In the network architecture C_m is context embedding of window size m, $h^{(1)}$ is the first hidden layer and $h^{(H)}$ is the last hidden layer in H hidden layer-sized network. In both the RNN architectures dotted arrow from $h^{(1)}$ to $h^{(H)}$ denotes the existence of multiple hidden connections between them. Similarly in Jordan network dotted arrow from softmax layer to hidden layer, represents the feeding of probability value to each hidden layer.
Note: Here the hypothetical real value vector of size 5 is used to demonstrate the network.

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 hyper-parameter values for Elman and Jordan model

training sample. Literature survey shows the performance improvements with the introduction of dropout. For both the RNN models, we set the value of dropout probability p as 0.5.

5.4 Results on Word Embedding Techniques

We have compared the impact of three word embedding techniques w.r.t Elman-type model as shown in Table-4. We have observed that CBOW outperform other two embedding models (RNNLM and Random Number) as it adapts distributional hypothesis while training. RNNLM obtained word vectors were very effective in capturing syntactic part because of its direct connection to the non-linear hidden layer. However, CBOW model was even better than RNNLM in identifying syntactic part and performs comparable on the semantic part.

Word Embedding Techniques	dimension (d)	precision	recall	F-measure
Random Number	100	94.19	85.48	89.62
RNNLM	80	94.21	87.98	90.98
CBOW	300	97.09	90.52	93.68

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

5.5 CRF Model: Baseline

Literature survey shows that majority of the existing systems on patient de-identification learn the CRF based classifier with features such as Chunk, Part-of-Speech (POS), n-gram character etc. This motivated us to develop supervised machine learning model based on CRF classifier as our baseline. The classifier is trained with a standard set of hand-crafted features, which are chosen based

on the best system of i2b2 2014 challenge (Yang and Garibaldi, 2015):

1. Context word feature: Local context plays very important role in identifying the current word. We use current word and the local context spanning from the preceding three to the succeeding three words.

2. Bag-of-word feature: We generated unigrams, bi-grams and tri-grams features within window size of $[-2, 2]$ w.r.t current word.

3. Part-of-Speech (PoS) Information: POS information is very helpful in identifying the entity as most of the entities belong to noun phrases. Here, we have generated features for current word, previous two words and next two words. We have used Stanford tagger (Toutanova and Manning, 2000) to extract POS information.

4. Chunk Information: In identification of boundary of PHI-term, chunk information plays a very important role. We have used Chunk information as feature from *openNLP*[4].

5. Combined POS-token and Chunk-token Feature: We have generated the combined feature of PoS and 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.

6. Task-specific Feature: A task-specific list is generated which includes all US states names and acronyms, names of countries, names of all days in a week, month, season, US festival. Apart from this we also include lexical clues w.r.t each PHI category such as "Ms.", "Mr." for patient, "Dr.", "M.D." in case of doctor.

7. Regular Expression Patterns: Specific regular expression patterns are designed for identifying PHI related information such as date, ID, age, phone number, username, medical record.

CRF based model was developed using abovementioned feature set. We performed experiments using the CRF implementation[5] of CRF^{++} using the default parameter. Table-5 provided the comprehensive results with the model build on CRF.

5.6 Results with Elman-RNN

We have implemented Elman RNN model as described in Subsection-4.3 to extract PHI terms from medical records. We have provided detailed evaluation results in Table-5 describing overall F-Measure as well as F-Measure value for every PHI categories separately. Obtained results shows that E-RNN outperforms CRF based model in identifying PHI terms. We have further evaluated E-RNN on different word embedding techniques as discussed in Subsection-5.4. We have obtained an interesting observation as shown in Table 4 that CBOW based word embedding outperforms other embedding technique when provided as input to E-RNN.

5.7 Results with Jordan-RNN

We have also implemented second variant of RNN, Jordan RNN for exploiting the effectiveness in identifying PHI terms. Jordan like Elman also outperforms the strong baseline model based on CRF. We present the detailed comparative results in Table-5. Obtained results show the effectiveness of J-RNN over the other two models. J-RNN performs better than E-RNN in identifying 5 PHI categories.

5.8 De-Identification of PHI terms

The final stage after identification of PHI terms is to de-identify those terms. It is required in order to preserve the medical contents of the records for their applicability in further research. A basic template is used to convert all the identified PHI terms, e.g., *Patient*, *Hospital*, *Doctor* etc. are converted into a generic format like XYZ_Patient, XYZ_Hospital, XYZ_Doctor respectively, and all the dates into the format 00_00_Date. Similarly, we also de-identify all the PHONE numbers and IDs by representing all the identified IDs and PHONE numbers as NUM_ID and NUM_PHONE, respectively. This helps to capture the information required without compromising the personal details.

6 Error Analysis

The results presented in Table-5 show the success of RNN model over the CRF-based baseline model. Detailed investigation of the output produced by the system yields the following:

(1) RNN model significantly fails in showing sustainable results in case of *ID* which is correctly identified by the CRF-based model due to the use of well-defined regular expression patterns.

(2) Inter-PHI ambiguity: These errors occur 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 name consists of

[4]https://opennlp.apache.org/

[5]https://taku910.github.io/crfpp/

PHI Category	CRF Model			Elman			Jordan		
	P	R	F	P	R	F	P	R	F
NAME	97.82	95.01	96.39	98.92	94.94	96.88	98.95	95.29	97.08
PROFESSION	74.24	70.25	72.18	81.01	75.25	78.02	81.94	75.93	78.82
LOCATION	85.47	86.28	85.87	94.74	88.98	91.76	94.24	89.57	91.84
AGE	96.18	92.28	94.18	97.92	92.89	95.33	98.81	92.17	95.37
DATE	98.25	94.96	96.57	98.64	93.47	95.98	98.95	94.98	96.92
CONTACT	97.86	94.23	96.01	97.25	95.91	96.57	97.84	93.12	95.42
ID	98.04	98.17	98.10	97.26	94.26	95.73	97.17	94.89	96.01
Micro-averaged	**94.89**	**89.28**	**91.99**	**97.09**	**90.52**	**93.68**	**97.26**	**90.67**	**93.84**

Table 5: Performance of CRF and RNN based models for identifying PHI at entity level. CRF is the baseline model based on Conditional Random Field. Elman and Jordan are two variants of RNN model. Our system is evaluated w.r.t *recall(R), precision (P)* and *F-measure (F)*. All the values are reported in %

Systems	Features & Rules	Techniques	External Resources	F-Measure
Our model		Deep Learning: RNN	Word vectors	93.84%
Nottingham (Yang and Garibaldi, 2015)	Regular Expression template for e.g. DATE, USERNAME, IDNUM, AGE, PHONE, MEDICAL RECORD	CRF: Sentence level, contextual, orthographic, word-token	Dictionary for US states, countries, week, month	93.60%
Harbin-Grad (Liu et al., 2015)	Regular Expression for FAX, MEDICAL RECORD , EMAIL, IPADDR, PHONE	CRF: Part of Speech (PoS), bag-of-words,affixes, orthographic features, dictionary feature, section information, word shapes		91.24%
Manchester (Dehghan et al., 2015)	Orthographic, contextual, entity, pattern	CRF: semantic, lexical, positional,orthographic	Wikipedia, DEID, GATE	90.65%
Harbin (He et al., 2015)	Regular expression patterns for tokenization	CRF: lexical, syntactic, orthographic		88.52%
Kaiser (Torii et al., 2014)	Regular expression patterns for EMAIL, PHONE, ZIP	Standford NER, no feature mentioned	De-ID corpus	81.83%
Newfoundland (Chen et al., 2015)		Bayesian HMM: token, number and word token		80.55%

Table 6: Comparisons with the existing systems. The F-measure value reported is on micro-averaged entity based evaluation.

single word. For examples, “Glass”, “Chabechird” etc.

(3) RNN models is seen to outperform CRF for detecting *PROFESSION* category. The main reason of RNN’s success is due to semantic and syntactic property captured by word embedding models.

(4) RNN model was able to capture the variations in the wordforms, which most of the time, is predicted incorrectly by a CRF-based model such as misspelling, tokenization and short wordform. For e.g., “KELLIHER CARE CENTER”, “KCC”, “20880703” etc.

(5) RNN models are able to capture semantic variance, which CRF model is unable to capture properly. The systems learned through RNN are trained on a large unlabeled corpus which makes RNN suitable in capturing the context efficiently which would be significantly time consuming for generating the features for every possible context.

(6) CRF model is seen to be good at identifying the words included in the dictionary or gazetteers, for e.g., “Christmas”. As “Christmas” never appears in the training set, RNN model fails to identify it. Whereas CRF identifies it properly because of its presence in the gazetteer list.

6.1 Discussion and Comparative Analysis

We have performed comprehensive study of two variants of RNN architectures, Elman and Jordan in identifying PHI terms. Both the RNN models outperform CRF based model which requires hand-crafted features. However, J-RNN was observed to be best model in identifying majority of the PHI categories. J-RNN adjusts the weights for current word considering output from both previous words and hidden layer not just from previous words unlike E-RNN. As a result of this, J-RNN was able to perform better on multi-word

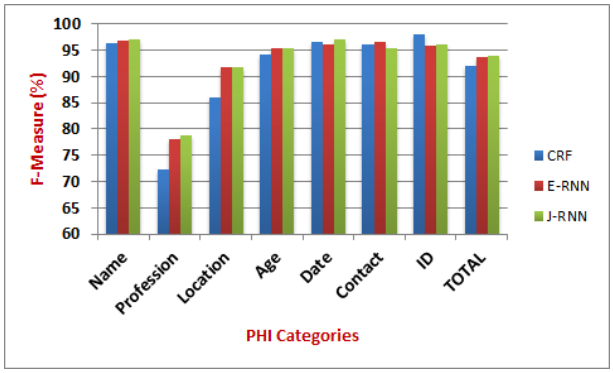


Figure 2: Comparison of CRF based model with Elman and Jordan models in term of F-Measures on 7 identified PHI categories

PHI terms[6]. We also compare with the state-of-art models as shown in Table-6. It shows that RNN model performs better compared to the machine learning based systems, including the best system of i2b2-2014 task (Yang and Garibaldi, 2015). Although the performance of our RNN based model is not tremendously high as compared to Nottingham system, it should be noted that their system was explicitly fine-tuned according to i2b2 dataset and evaluation framework. They performed post-processing on the identified PHI tokens. For e.g., changing "3041023MARY" to "304102" and "MARY", for term "MWFS" to "M", "W", "F", "S".

7 Conclusions

This paper presents the application of deep neural network architecture for solving de-identification task that is designed to identify and classify Protected Health Information (PHI) present in free-text medical records. We have systematically compared different variants of RNN architectures, including Elman, Jordan. We have also explored the effectiveness of using the word embedding for de-identification task. We observed the significant improvement of RNN type model over CRF based baseline. Experiments on the benchmark datasets over the baseline show the performance improvement of 1.69% and 1.85% with the Elman-type and Jordan-type network respectively. RNN based techniques also significantly outperforms the existing state-of-art systems. Future work will explore other effective learning methods for RNN such as Long Short term Memory (LSTM) as well exploring some other word embedding technique. We would also like to perform experiments with word embedding trained on clinical data.

[6]In multi-word NE, previous label provide effective information to predict the current word.

References

Eiji Aramaki, Takeshi Imai, Kengo Miyo, and Kazuhiko Ohe. 2006. Automatic deidentification by using sentence features and label consistency. In *i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data*, pages 10–11.

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

Tao Chen, Richard M Cullen, and Marshall Godwin. 2015. Hidden markov model using dirichlet process for de-identification. *Journal of Biomedical Informatics*, 58:S60–S66.

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.

Pascal Coorevits, M Sundgren, GO Klein, A Bahr, B Claerhout, C Daniel, M Dugas, D Dupont, A Schmidt, P Singleton, et al. 2013. Electronic health records: new opportunities for clinical research. *Journal of internal medicine*, 274(6):547–560.

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

Azad Dehghan, Aleksandar Kovacevic, George Karystianis, John A Keane, and Goran Nenadic. 2015. Combining knowledge-and data-driven methods for de-identification of clinical narratives. *Journal of biomedical informatics*, 58:S53–S59.

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

R Guillen. 2006. Automated de-identification and categorization of medical records. In *i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data*, volume 116.

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.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Zengjian Liu, Yangxin Chen, Buzhou Tang, Xiaolong Wang, Qingcai Chen, Haodi Li, Jingfeng Wang, Qiwen Deng, and Suisong Zhu. 2015. Automatic de-identification of electronic medical records using token-level and character-level conditional random fields. *Journal of Biomedical Informatics*, 58:S47–S52.

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.

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*.

Lawrence Rabiner and B Juang. 1986. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16.

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.

Shweta, Asif Ekbal, Sriparna Saha, and Pushpak Bhattacharyya. 2016. Deep learning architecture for patient data de-identification in clinical records. In *Proceeding of Clinical Natural Language Processing Workshop (ClinicalNLP) at the 26th International Conference on Computational Linguistics (COLING 2016), Japan (accepted)*.

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.

Manabu Torii, Jung-wei Fan, Wei-li Yang, Theodore Lee, Matthew T Wiley, Daniel Zisook, and Yang Huang. 2014. De-identification and risk factor detection in medical records. In *Seventh i2b2 Shared Task and Workshop: Challenges in Natural Language Processing for Clinical Data*.

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.

Özlem Uzuner, Yuan Luo, and Peter Szolovits. 2007. Evaluating the state-of-the-art in automatic de-identification. *Journal of the American Medical Informatics Association*, 14(5):550–563.

Ben Wellner, Matt Huyck, Scott Mardis, John Aberdeen, Alex Morgan, Leonid Peshkin, Alex Yeh, Janet Hitzeman, and Lynette Hirschman. 2007. Rapidly retargetable approaches to de-identification in medical records. *Journal of the American Medical Informatics Association*, 14(5):564–573.

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