Journal of Biomedical Informatics 75 (2017) S34–S42

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

De-identification of clinical notes via recurrent neural network and conditional random field

Zengjian Liu, Buzhou Tang *, Xiaolong Wang, Qingcai Chen

Key Laboratory of Network Oriented Intelligent Computation, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China

ARTICLE INFO

Article history:
Received 29 January 2017
Revised 26 May 2017
Accepted 30 May 2017
Available online 1 June 2017

Keywords:
De-identification
Protected health information
Natural language processing
Recurrent neural network
Ensemble system

ABSTRACT

De-identification, identifying information from data, such as protected health information (PHI) present in clinical data, is a critical step to enable data to be shared or published. The 2016 Centers of Excellence in Genomic Science (CEGS) Neuropsychiatric Genome-scale and RDOC Individualized Domains (N-GRID) clinical natural language processing (NLP) challenge contains a de-identification track in de-identifying electronic medical records (EMRs) (i.e., track 1). The challenge organizers provide 1000 annotated mental health records for this track, 600 out of which are used as a training set and 400 as a test set. We develop a hybrid system for the de-identification task on the training set. Firstly, four individual subsystems, that is, a subsystem based on bidirectional LSTM (long-short term memory, a variant of recurrent neural network), a subsystem-based on bidirectional LSTM with features, a subsystem based on conditional random field (CRF) and a rule-based subsystem, are used to identify PHI instances. Then, an ensemble learning-based classifiers is deployed to combine all PHI instances predicted by above three machine learning-based subsystems. Finally, the results of the ensemble learning-based classifier and the rule-based subsystem are merged together. Experiments conducted on the official test set show that our system achieves the highest micro F1-scores of 93.07%, 91.43% and 95.23% under the "token", "strict" and "binary token" criteria respectively, ranking first in the 2016 CEGS N-GRID NLP challenge. In addition, on the dataset of 2014 i2b2 NLP challenge, our system achieves the highest micro F1-scores of 96.98%, 95.11% and 98.28% under the "token", "strict" and "binary token" criteria respectively, outperforming other state-of-the-art systems. All these experiments prove the effectiveness of our proposed method.

1. Introduction

Nowadays, there have been billions of electronic medical records (EMRs), which are very important for data-driven medical research and production. However, companies, organizations and researchers are not free to use EMRs because of a large amount of protected health information (PHI) embedded in them protected by Health Insurance Portability and Accountability Act (HIPAA) [1]. It is not allowed to use EMRs with PHI instances until all of them are de-identified. Therefore, de-identification that identifies and removes PHI is a primary step in making clinical data accessible to more people and attracts great attention. In the past few years, lots of efforts had been made for de-identification. The representative works are three natural language processing (NLP) challenges, two organized by the Center of Informatics for Integrating Biology and Bedside (i2b2) in 2006 [2] and 2014 [3–5], and one organized by the Centers of Excellence in Genomic Science (CEGS) Neuropsychiatric Genome-scale and RDOC Individualized Domains (N-GRID) in 2016 [6]. The organizers of the three challenges provide manually annotated corpora for participants to develop various kinds of systems for de-identification [7–15].

De-identification is one of the three tasks of the 2016 CEGS N-GRID NLP challenge. We participate in the CEGS N-GRID NLP challenge and develop an ensemble system for the de-identification task on the training set. In our system, an ensemble classifier is deployed to combine the outputs of three individual machine learning-based subsystems, and a rule-based subsystem is used to identify some formulaic PHI instances. The three machine learning-based subsystems are a CRF-based system with a large number of hand-crafted features [12], a bidirectional LSTM-based system without any hand-crafted features [16,17], and a variant of bidirectional LSTM-based system with a small quantity of hand-crafted features [18,19]. Moreover, we also evaluate our system on the 2014 i2b2 challenge corpus and compare it with other state-of-the-art systems. In the remainder of this paper, we give a brief introduction to background in Section 2, depict methods and

* Corresponding author.
E-mail addresses: liuzengjian.hit@gmail.com (Z. Liu), tangbuzhou@gmail.com (B. Tang), wangxl@insun.hit.edu.cn (X. Wang), qingcai.chen@gmail.com (Q. Chen).

http://dx.doi.org/10.1016/j.jbi.2017.05.023
1532-0464/

materials in detail in Section 3, present and analyze experimental results in Sections 4 and 5, and draw conclusions in Section 6.

2. Background

In the last decades, various NLP approaches have been proposed for de-identification, which is a typical named entity recognition (NER) problem. The early de-identification systems in the clinical domain are mainly rule-based, such as Sweeney et al.'s [20], Gupta et al.'s [21], etc. They employed a large number of rules, patterns and specialized semantic dictionaries to identify PHI instances in different types of EMRs, such as pathology reports [21–23], laboratory reports [24], X-ray reports [25], discharge summaries [24,25], etc. The main problem of the rule-based systems lies in that the rules in one system are not easily applicable to another system. In addition, it is not easy to evaluate different systems, as there is no publicly available annotated dataset.

To accelerate research on de-identification in the clinical domain, three related challenges have been organized, that is, the 2006 i2b2 NLP challenge, the 2014 i2b2 NLP challenge, and the 2016 CEGS N-GRID NLP challenge. Lots of teams from all around the world participated in this three challenges. In the two i2b2 NLP challenges, the proposed de-identification systems may fall in three categories: rule-based [26], machine learning-based [10,14,27], and hybrid [11–13,15]. The rule-based systems can exactly recognize formulaic PHI instances (i.e., phone numbers, emails, licenses, etc.), but need more complex rules and dictionaries to extract diverse PHI instances such as names, professions, hospitals, etc. The machine learning-based methods can perform well on diverse PHI instance recognition when we have sufficient training samples and can obtain rich features. However, they are not good at recognizing complex formulaic PHI instances such as emails with complex compositions. The hybrid methods take full advantage of the rule-based and machine learning-based methods by combining results of them, and usually achieve better performance than each category of them. Results of the 2014 i2b2 NLP challenge also demonstrated that the hybrid systems outperformed the other two categories of systems. The machine learning algorithms used in these systems included conditional random field (CRF) [28], structured support vector machine (SSVM) [29], support vector machine (SVM) [30], hidden Markov model (HMM) [31], decision tree (DT) [32], and so on. Some of them, such as CRF, SSVM and HMM, considered de-identification as a sequence labeling problem and the others, such as SVM and DT, modeled de-identification as a classification problem. The machine learning algorithms for sequence labeling usually outperform the algorithms for classification as they take advantage of interactions between neighbor labels. For de-identification, CRF is the most popular machine learning algorithm used. For example, the top four systems of the 2014 i2b2 de-identification challenge were all based on CRF. One of key points for machine learning-based systems is feature engineering, which is task-specific and hand-crafted. The features used for de-identification include N-grams, part-of-speech (POS), word vector features [12,33], dictionary features [11–13], etc. Beside hand-crafted features, pre-processing and post-processing certainly affects the performance of machine learning-based systems. Researchers also contributed in pre-processing and post-processing. For example, Liu et al. (2015) [12] proposed a character-level tokenization method to avoid some errors caused by existing tools. Yang et al. (2015) [13] uncovered more potential PHI instances by maintaining a trusted PHI instance list at the post-processing phase.

In recent years, recurrent neural network (RNN) [34] has been widely used to tackle various kinds of NLP tasks, such as machine translation [35], NER [16–19,36,37], word sense disambiguation [38], syntax parsing [39,40], etc., and has shown great potential. For NER, there have already been several popular neural architectures, such as Senna (Collobert et al., 2011) [37], CharWNN (dos Santos et al., 2015) [36], LSTM-CRF (Huang et al., 2015 [19] and Lample et al., 2016 [17]), LSTM-CNNs-CRF (Ma et al., 2016) [17], and LSTM-CNNs (Chiu et al., 2016) [18]. Collobert et al. (2011) [37] proposed a feed-forward neural network model with capitalization and discrete suffix features. dos Santos et al. (2015) [36] used a neural network model based on Senna with character-level word representations modeled by convolutional neural network (CNN). Huang et al. (2015) [19] presented a bidirectional LSTM model and utilized discrete spelling, POS and context features. Ma et al. (2016) [16] used a bidirectional LSTM model with character-level word representations modeled by CNN. Chiu et al. [18] used the similar architecture to Ma et al. (2016) [16] and further introduced some features such as the character type, capitalization, and lexicon features. Instead of CNN for character-level word representations, Lample et al. (2016) [17] adopted LSTM to model character-level word representations. Lots of experiments demonstrated that bidirectional LSTM and character-level representation brought great improvements for sequence labeling problems, such as name entity recognition, some additional features were also beneficial. This brings a great inspiration for de-identification.

Recently, RNN have also been used for de-identification of clinical notes, and showed better performance than CRF. For example Dernoncourt et al. (2016) [41] deployed a bidirectional LSTM model similar to Lample et al.'s (2016) [17] to de-identification. In this study, we compared several individual methods for de-identification, including CRF, bidirectional LSTM, bidirectional LSTM with some hand-crafted features and a rule-based method, and proposed an ensemble method to combine the results of them. In both LSTM-based models, we deploy character-level representation since it can capture the morphological information of words and has been proved useful for NER including de-identification [17,41].

3. Material and methods

Fig. 1 shows the overview architecture of our system for de-identification task. Firstly, we tokenize raw clinical texts, then deploy four individual methods (CRF-based, bidirectional LSTM, bidirectional LSTM with features and Rule-based methods) for de-identification respectively, and use an ensemble learning method to combine all PHI instances predicted by the first three machine learning-based methods, finally merge all results of the ensemble classifier and rule-based method together. The detailed description of our system is presented below.

3.1. Dataset

Two benchmark corpora are used to evaluate our system: 2014 i2b2 and 2016 N-GRID. The 2014 i2b2 dataset was released as part of track 1 of the 2014 i2b2 NLP challenge [5], and the 2016 N-GRID dataset as part of track 1 of the 2016 CEGS N-GRID NLP challenge [6]. They are the latest publicly available datasets for de-identification. All PHI instances in seven main categories and twenty-seven subcategories (including all HIPAA-defined PHI categories) are annotated according to the same guidelines. The 2014 i2b2 corpus is composed of a training set of 790 records with 17,045 PHI instances and a test set of 514 records with 11,462 PHI instances. The 2016 N-GRID corpus is composed of a training set of 600 records with 20,845 PHI instances and a test set of 400 records with 13,519 PHI instances. The numbers of PHI instances of main categories in the two corpora are listed in Table 1, where

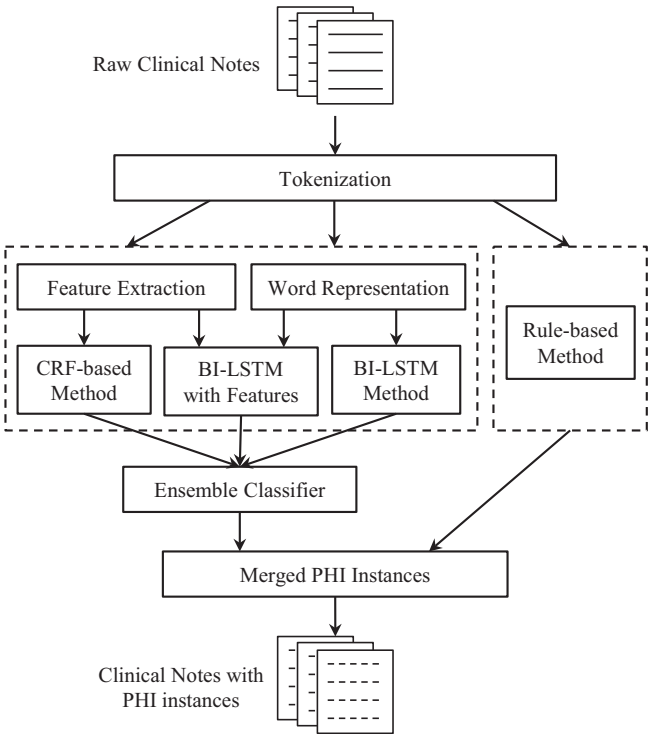

Fig. 1. Overview architecture of our de-identification system.

Table 1
Numbers of PHI instances of main categories in the full 2014 i2b2 and 2016 N-GRID corpora.

Main category	Sub category	2014 i2b2	2016 N-GRID
NAME	PATIENT∗, DOCTOR USERNAME	7348	6095
PROFESSION	NA	413	2,481
LOCATION	COUNTRY, STATE, CITY∗ STREET∗, ZIP∗, HOSPITAL ORGANIZATION∗, ROOM DEPARTMENT, OTHER	4580	9896
AGE∗	NA	1,997	5,991
DATE∗	NA	12,482	9,544
CONTACT	PHONE∗, FAX∗, EMAIL∗ URL, IPADDR	541	280
ID	MEDICALRECORD∗, SSN∗ DEVICE∗, IDNUM∗, BIOID∗ HEALTHPLAN∗, VEHICLE∗ ACCOUNT∗, LICENSE∗	1506	77
Total	NA	28,867	34,364

"NA" denotes no subcategory, and asterisks denote the HIPAA-defined categories.

3.2. Tokenization

Tokenization is a very important preprocessing step for NLP, which splits sentences into a sequence of tokens, the smallest component of PHI instances. Character-level tokenization and existing tokenization tool have been used in de-identification systems submitted to the 2014 i2b2 NLP challenge [12]. Both of them have their own shortcomings. The character-level tokenization, which decomposes all sentences into characters directly, broke normal words and numbers into single characters, for example, "71Total" was broken into "7 1 T o t a l". The existing tokenization tool brought some unexpected errors such as "CBT.", where the full stop characters '.' can't be split from abbreviations "CBT" as they were confused with the format of special abbreviations ending with dot such as "Phys.". Therefore, in our system for the 2016 CEGS N-GRID NLP challenge, we design a new tokenization module. Firstly, we split sentences into tokens by blank spaces, then further separate consecutive numbers, consecutive letters and other characters. For example, sentence "1/20/71Total time of visit (in minutes):" is tokenized into "1", "/", "20", "/", "71", "Total", "time", "of", "visit", "(", "in", "minutes", ")", ":". This proposed tokenization module could preserve as many normal words and numbers as possible while avoiding errors caused by the existing tokenization tools. Experiments conducted on the corpus of the 2014 i2b2 NLP challenge by us proved that the new tokenization module is better than the previous two tokenization modules (i.e., character-level tokenization and existing tokenization tools).

3.3. CRF-based method

As previous work in [12], we proposed a CRF-based method for de-identification with above mentioned tokenization module. The features used in this method include:

Bag-of-words: unigrams, bigrams and trigrams of words within a window of [-2, 2].
Part-of-speech (POS) tags: unigrams, bigrams and trigrams of POS tags within a window of [−2, 2]. The Stanford POS Tagger [42] was used for POS tagging.
Combinations of words and POS tags: combining current word with the unigrams, bigrams and trigrams of POS tags within a window of [−1, 1], i.e. w_0p_{-1}, w_0p_0, w_0p_1, $w_0p_{-1}p_0$, $w_0p_0p_1$, $w_0p_{-1}p_1$, $w_0p_{-1}p_0p_1$, where w_0, p_{-1}, p_0 and p_1 denote current word, last, current and next POS tags respectively.
Sentence information: number of words in current sentence, whether there is an end mark at the end of current sentence such as '.', '?' and '!', whether there is any bracket unmatched in current sentence.
Affixes: prefixes and suffixes of length from 1 to 5.
Orthographical features: whether the word is upper case, contains uppercase characters, contains punctuation marks, contains digits, etc.
Word shapes: mapping any or consecutive uppercase character (s), lowercase character(s), digit(s) and other character(s) in current word to ' A', ' a', ' #' and ' -' respectively. For instance, the word shapes of "Hospital" are "Aaaaaaaa" and "Aa".
Section information: twenty-nine section headers (see the supplementary file) were collected manually such as "History of Present Illness"; we check which section current word belongs to.
General NER information: the Stanford Named Entity Recognizer [43] was used to generate the NER tags of current word, include: person, date, organization, location, and number tags, etc.
Word representation features: two types of word representation features generated by Brown clustering [44] and word2vec [45] on training sets and a large unlabeled MEDLINE corpus.
Dictionary features: four categories of localization dictionaries: COUNTRY, STATE, CITY and ZIP were collected from Internet, and each word was labeled with ' 0' or ' 1' by dictionary lookup.

We use CRFsuite [46] as the implementation of CRFs. A CRF-based de-identification system, developed on the de-identification corpus of the 2014 i2b2 NLP challenge is available at: http://icrc.hitsz.edu.cn/Article/show/144.html.

3.4. Bidirectional LSTM (BI-LSTM)

Bidirectional LSTM is a deep learning method for sequence labeling problem. As shown in Fig. 2 (suppose that PHI instances are represented by "BIO" tags), it contains three main layers: (1) input layer, which generates the representation of each word in a sentence, and contains two parts: character-level representation (denoted by grey squares) and token-level representation (denoted by blank squares); (2) LSTM layer, which includes a forward LSTM and a backward LSTM, takes the word representation sequence of a sentence as input, and outputs a new word representation sequence that captures the context information of each word in this sentence; (3) CRF layer, which captures the dependencies between successive labels by keeping a label transition matrix referring to the conditional random field (CRF) algorithm, and predicts the best label sequences with correct structures. The architecture of our BI-LSTM is the same as Lample et al.'s (2016) architecture for NER [17]. We will introduce these three layers in detail in the following sections.

3.4.1. Input layer

Two different types of word representations were used in this study: token-level and character-level, which capture context information and morphological information of words respectively. The token-level word representation was generated by looking up a pre-trained word embedding, such as SENNA [37,47], trained by neural language models, such as continuous bag-of-words (CBOW) and skip-gram [45], on a large unlabeled data. The character-level word representation was directly learned from the character sequence of each word by a bidirectional LSTM as shown in Fig. 2 (i.e., Joan), which may capture both the prefix and suffix information of each word. In this bidirectional LSTM, the last two vectors of the forward and backward LSTMs are concatenated together to form the final character-level representation of the word (i.e. Joan).

3.4.2. LSTM layer

Give a sentence $s = w_1 w_2 \dots w_n$ with each word w_t $(1 \leqslant t \leqslant n)$ represented by x_t, the LSTM layer takes a word representation sequence $x = x_1 x_2, \dots, x_n$ as input and produces a new word representation sequence $h = h_1 h_2, \dots, h_n$, where $h_t = \left[h_{ft}^T, h_{bt}^T\right]^T$ $(1 \leqslant t \leqslant n)$ is a concatenation of the outputs of both forward LSTM h_{ft} and backward LSTM h_{bt} at step t. More specifically, an LSTM unit (composed of an input gate, a forget gate, an output gate and a memory cell) at step t that takes x_t, h_{t-1} and c_{t-1} as input and produces h_t and c_t via the following formulas:

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
\tag{1}
$$

where σ is the element-wise sigmoid function; $\odot$ is the element-wise product; i_t, f_t and o_t are input, forget, and output gates; c_t is the cell vector; W_i, W_f, W_c, W_o (with subscripts: x, h and c) are the weight matrices for input x_t, hidden state h_t and memory cell c_t respectively; b_i, b_f, b_c and b_o denote the bias vectors.

3.4.3. CRF layer

The CRF layer takes sequence $h = h_1 h_2 \dots h_n$ as input, and outputs the most possible label sequence $y = y_1 y_2, \dots, y_n$. Given a training set D, all parameters of CRF (denoted as θ) are estimated by maximizing the following log-likelihood:

$$
L(\theta) = \sum_{(s,y)\in D} \log p(y|h,\theta) \tag{2}
$$

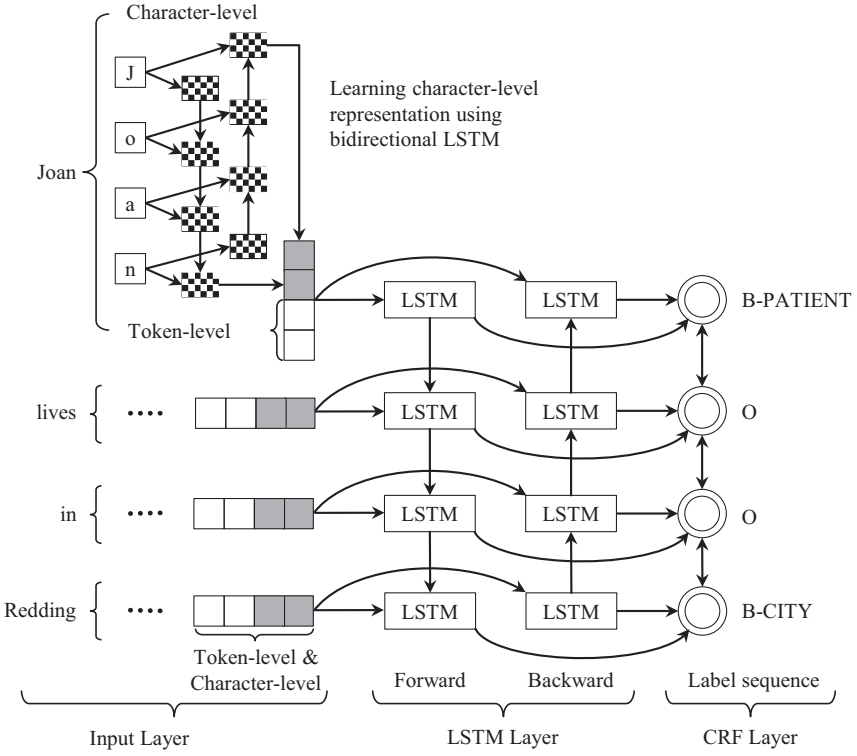

Fig. 2. Overview architecture of BI-LSTM.

where y is the corresponding label sequence of sentence s, h is the word representation sequence for sentence s outputted by the network, and p is the conditional probability of y when given s and θ. Assuming that $Z_\theta(h,y)$ is the score of label sequence y for sentence h, the conditional probability p can be calculated by:

$$p(y|h,\theta) = \frac{e^{Z_\theta(h,y)}}{\sum_{y'} e^{Z_\theta(h,y')}} \quad (3)$$

where y' is a possible label sequence of h. The log-likelihood of p is:

$$\log p(y|h,\theta) = Z_\theta(h,y) - \log \Sigma_{y'} e^{Z_\theta(h,y')} \quad (4)$$

In order to model dependencies between neighbor labels, a transition matrix T is incorporated with a emission matrix E to $Z_\theta(h,y)$ as follow:

$$Z_\theta(h,y) = \sum_{t=1}^{n} (E_{y_t,t} + T_{y_{t-1},y_t}) \quad (5)$$

where $E_{y_t,t}$ is the probability that token h_t with label y_t, and T_{y_{t-1},y_t} is the probability that token h_{t-1} with label y_{t-1} followed by h_t with label y_t. We can maximize the log-likelihood (2) over all training set D by the dynamic programing, and find the best label sequence for each sentence by maximizing score (5) using Viterbi algorithm at test phase.

3.5. BI-LSTM with Features (BI-LSTM-FEA)

Similar to [18,19], we introduce distributed representations for features and extend BI-LSTM by adding a hidden layer after the LSTM layer, which concatenates the word representations generated by LSTM layer and the feature representations, as shown in Fig. 3. The features used in BI-LSTM-FEA are: sentence information, section information, general NER information and dictionary features, which are the same as features mentioned in section "CRF-based Method".

3.6. Rule-based method

Since PHI instances in some categories are formulaic and their amounts are very small, we manually define some specific rules to extract them directly, and do not use machine learning-based methods to extract them anymore. The detailed regular expressions for these categories of PHI instances are listed in Table 2. It should be stated here that we only apply the rule-based method on the 2016 N-GRID corpus but not on the 2014 i2b2 corpus as PHI instances in these categories in the 2014 i2b2 corpus are much more numerous than in the 2016 N-GRID corpus.

3.7. Ensemble learning method

To take full advantages of above individual machine learning-based methods, we use an ensemble learning method [48], a support vector machine (SVM) classifier, to merge all PHI instances predicted by them. We use LibSVM [49] as an implementation of SVM. The goal of the ensemble learning method is to determine whether a predicted PHI instance from all above methods is a true instance, and the features used in this SVM-based classifier are:

- Whether the text spans of a PHI instance exactly match with others?
- Whether the text spans exactly match with others of the same type?
- Whether the text spans of a PHI instance partially match with others?
- Whether the text spans partially match with others of the same type?
- Whether the text of a PHI instance contains a conjunction or preposition?
- Which methods have predicted this PHI instance?
- How many times a PHI instance was predicted?
- How many times the span of a PHI instance was predicted?
- The number of tokens in a PHI instances.
- How many times a PHI instance was predicted in a same clinical record?

3.8. Evaluation

All our evaluations are performed on the official test sets using the evaluation tool provided by the organizers of 2016 CEGS N-GRID NLP challenge. The tool outputs micro-average precisions (P), recalls (R), and F1-scores (F1) under ten criteria, which are introduced in the overview paper for the 2016 CEGS N-GRID NLP de-identification challenge [6]. They are "token", "strict", "relaxed", "HIPAA token", "HIPAA strict", "HIPAA relaxed", "binary token", "binary strict", "binary HIPAA token" and "binary HIPAA strict", where "token" checks whether a predicted token exactly matches a token in a gold phrase of the same category, "strict" denotes that a PHI instance is correctly extracted only when it exactly matches with a gold one of the same boundary and category, "relaxed" denotes that a PHI instance is correctly extracted only when it mostly overlaps (only allow two characters mismatched at the end) with a gold one of the same category, "HIPAA" only considers nineteen types of HIPAA-defined PHI instances, and "binary" only considers the boundaries of PHI instances no matter

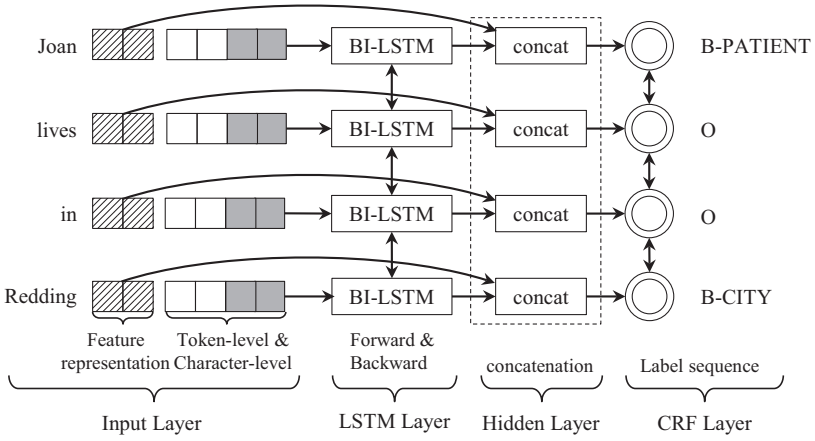

Fig. 3. Overview architecture of BI-LSTM-FEA.

Table 2
Regular expressions used for some categories.

PHI Category	Regular Expression	Remark
PHONE	\(?\d{3}\)?[- \t]?\d{3}[- \t]\d{4} \d{3}\.\d{3}\.\d{4}	(304) 911–4864 017.445.0833
FAX	[Ff]ax(ed)?[^\d\n]{0,20}(\(?\d{3}\)? [- \t]?\d{3}[- \t]\d{4})	Fax: 518–893-5533
EMAIL	\w+@\w+(\.\w+)*\. (com|net|edu|org)	hcuutaj@bdd.com
URL	((www\.\w+(\.\w+)+)|((\w + \.) +com)) [\.\)\"\'\]\}\>,;]?\s	Pills4U.com
LICENSE	[A-Z]{2}[-]?[ ]?\d{7}	LQ 7477948

their categories. Among these ten criteria, "strict" is the primary one.

3.9. Experiments setup

In this study, we use "BIO" (B-beginning of an instance, I-insider an instance, O-outsider an instance) to represent the PHI instances, and use the stochastic gradient descent (SGD) algorithm for parameter estimation of neural network models with the same hyper-parameters (all shown in Table 3 except "Training epochs") used by Lample et al. (2016) for named entity recognition [17]. The number of training epochs on each corpus is determined on an independent development set split from the training set. Finally, the numbers of training epochs on both corpora are set to eighty. Therefore, all neural network models have the same hyper-parameters shown in Table 3. The pre-trained word embedding SENNA is used to initialize the token-level word representations, and the character-level word representations and feature representations are randomly initialized from a uniform distribution ranging in [−1, 1]. The two types of word representations and the feature representations are fine-tuned during training. All experiments are performed on a Linux server with four CPUs (6 cores), and all models are evaluated on the official test sets. The training times of CRF, BI-LSTM and BI-LSTM-FEA models are about 8, 57 and 60 h respectively on the 2014 i2b2 training set, while 10, 94 and 95 h on the 2016 N-GRID training set. Training the ensemble classifier is completed within about two minutes on both training sets.

4. Results

The "strict" micro-average precisions, recalls and F1-scores of our system and its subsystems (shown in Fig. 1) are listed in Table 4 (full results in supplementary file), where "NA" denotes no result. Among the three machine learning-based subsystems, the two subsystems based on LSTM (i.e., BI-LSTM and BI-LSTM-FEA) achieve much better F1-scores than the CRF-based subsystem, and BI-LSTM-FEA achieves a little better F1-score than BI-LSTM with "strict" micro-average F1-scores of 94.51% on the 2014 i2b2 and 89.98% on the 2016 N-GRID corpora, higher than BI-LSTM and CRF-based by 0.22% and 1.93% on the 2014 i2b2 corpus, 0.86% and 1.61% on the 2016 N-GRID respectively. The ensemble classifier based on the three machine learning-based subsystems outperformed the best individual one (i.e., BI-LSTM-FEA) by "strict" micro-average F1-scores of 0.60% on the 2014 i2b2 corpus and 0.94% on the 2016 N-GRID corpus. When the ensemble classifier and the rule-based subsystem merged together, the final system ("Overall") achieves highest "strict" micro-averaged F1-scores of 91.43% on the 2016 N-GRID corpus. As reported by the 2016 CEGS N-GRID NLP challenge organizers [6], our system ranks first in the challenge. On the two corpora, all systems perform better on the 2014 i2b2, and the "strict" F1-score difference of the best system is 3.68%.

Table 3
Hyper-parameters used for the neural network.

Hyper-parameter	Value
Dimension of token-level word representation	50
Dimension of character word representation	25
Dimension of feature representation	25
Character-level LSTM size	25
Token-level LSTM size	100
Dropout probability	0.5
Learning rate	0.005
Training epochs	80

In order to investigate on which category of PHI instances our model performs well, we list the performances of our de-identification system on main categories of PHI instances under "strict" criterion in Table 5. We can see that our system performs well on most categories of PHI instances, but not very well on some categories such as "PROFESSION" (75.82% in F1-score) and "LOCATION" (88.67% in F1-score) on the 2014 i2b2 corpus, "PROFESSION" (77.40% in F1-score), "LOCATION" (84.92% in F1-score) and "ID" (65.45% in F1-score) on the 2016 N-GRID corpus.

For convenient comparison with related studies and easy reference for other future works, we list the best results of our de-identification system on both 2014 i2b2 and 2016 N-GRID corpora in Table 6 under all the ten criteria. The highest "strict" F1-scores of our system for HIPAA-defined PHI instances are 96.75% on the 2014 i2b2 corpus and 92.93% on the 2016 N-GRID corpus. On the 2014 i2b2 corpus, our proposed system outperforms the best system of the 2014 i2b2 NLP challenge [13] (95.11% vs. 93.60% in "strict" micro-averaged F1-score) and the LSTM-based system recently proposed by Dernoncourt et al. [41] (98.27% vs. 97.88% in "HIPAA binary token" micro-averaged F1-score).

5. Discussion

As shown in Table 4, the reason why the LSTM-based subsystems are better than the CRF-based subsystem lies in that the LSTM-based subsystems achieve much higher recalls (93.34% and 93.61% vs. 90.12% on 2014 i2b2, 88.21% and 88.57% vs. 84.61% on 2016 N-GRID), indicating that LSTM is able to identify more PHI instances than CRF, which is similar to the related studies such as [41]. For instance, "KC" is a patient name appearing in one record more than thirty times in form of "KC reports…", "KC states…", "KC denies…", "…have told KC…", etc., as shown in Table 7. The CRF-based subsystem only recognizes ten of them, while the LSTM-based subsystems recognize all of them. "NPH", appearing in four sentences in another documentas shown in Table 7, are successfully identified as HOSPITAL instances by LSTM-based subsystems, but not by CRF-based subsystem. The reason may lie in that the LSTM-based method was able to generalize the contexts of "KC" and "NPH" better than CRF-based method, through a better similarity of the series of context words, such as "reports", "states", "denies", and "presents to", "prior to", "discuss with".

When a small number of hand-crafted features are added to BI-LSTM (from BI-LSTM to BI-LSTM-FEA), both precision and recall are improved. It means that proper feature engineering is a valuable supplement to LSTM for de-identification. In our study, we select features beyond context. For example, words not appearing training sets, such as "Rockville", "Harlingen", "Austria", "Bosnia", etc., can be correctly recognized as countries or cities by BI-LSTM-FEA but not by BI-LSTM as they can be found in a gazetteer or identified as countries or cities by a NER tool.

Table 4
Results of various methods ("strict", %).

Methods	2014 i2b2			2016 N-GRID		
	Precision	Recall	F1-score	Precision	Recall	F1-score
CRF-based	95.16	90.12	92.58	92.47	84.61	88.37
BI-LSTM	95.26	93.34	94.29	90.05	88.21	89.12
BI-LSTM-FEA	95.43	93.61	94.51	91.43	88.57	89.98
Ensemble	96.46	93.80	95.11	94.05	88.00	90.92
Rule-based	NA	NA	NA	91.10	0.984	1.947
Overall	96.46	93.80	95.11	94.22	88.81	91.43

Table 5
Performances of our de-identification system on main categories of PHI instances ("strict", %).

Main category	2014 i2b2			2016 N-GRID		
	Precision	Recall	F1-score	Precision	Recall	F1-score
NAME	95.42	94.03	94.72	96.20	91.60	93.84
PROFESSION	91.34	64.80	75.82	85.61	70.69	77.44
LOCATION	92.66	85.00	88.67	89.87	80.48	84.92
AGE	98.66	96.34	97.48	97.36	95.67	96.51
DATE	98.34	97.69	98.02	96.99	96.05	96.52
CONTACT	97.65	95.41	96.52	92.74	91.27	92.00
ID	94.41	91.84	93.11	81.82	54.55	65.45

Table 6
Best results of our de-identification system.

Criterion	2014 i2b2			2016 N-GRID		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Token	97.94	96.04	96.98	95.56	90.70	93.07
Strict	96.46	93.80	95.11	94.22	88.81	91.43
Relaxed	96.62	93.95	95.26	94.39	88.97	91.60
Binary token	99.30	97.28	98.28	97.78	92.81	95.23
Binary strict	97.90	95.15	96.50	95.97	90.47	93.14
HIPAA token	98.73	97.52	98.12	96.39	92.41	94.36
HIPAA strict	97.60	95.92	96.75	95.22	90.75	92.93
HIPAA relaxed	97.74	96.06	96.89	95.41	90.94	93.12
HIPAA binary token	98.88	97.66	98.27	97.06	93.05	95.01
HIPAA binary strict	97.76	96.07	96.91	95.70	91.21	93.40

Table 7
Examples of PHI instances (in bold) identified by the LSTM but not by the CRF.

PHI category	Examples
NAME	**KC** reports things start to be problematic 6 mo ago. **KC** states that he has had periods of sobriety... ...yet **KC** denies feelings of depression. ...people has told **KC** they think he' depressed...
HOSPITAL	...presents to **NPH** for psychopharm evaluation. With long wait to see auboxone provider in **NPH**. ...using cocaine prior to **NPH** tox screen... Will discuss with **NPH** team.

Although BI-LSTM-FEA achieves higher F1-scores than BI-LSTM, and BI-LSTM achieves higher F1-scores than CRF on the two corpora, each of them identifies certain PHI instances that cannot be identified by other two methods. According to our statistics, the numbers of true positive (TP) and false positive (FP) PHI instances only identified by each individual method on the 2016 N-GRID corpus are: (174 and 486) for CRF, (230 and 694) for BI-LSTM, and (181 and 474) for BI-LSTM-FEA. For this reason, we try to ensemble them together for further improvement. Actually, the ensemble classifier keeps 161 TP and 120 FP PHI instances that only identified by any individual method on the N-GRID corpus, which makes us obtain higher F1-scores.

In addition, we also investigate the performance of different methods (i.e., CRF, BI-LSTM, BI-LSTM-FEA and Ensemble as shown in Table 4) on each main PHI category (as shown in Fig. 4). No method outperforms the other ones on all PHI categories. For example, compared with CRF, BI-LSTM achieves higher "strict" F-scores on all PHI categories except ID on the 2014 i2b2 corpus and LOCATION on the 2016 N-GRID corpus. Even the Ensemble method achieves a slightly lower "strict" F1-score than BI-LSTM-FEA on PROFESSION PHI instances on the 2014 i2b2 corpus. The reason may lie in that when we introduce new information, some errors may be fixed, but a few of new errors may be brought.

As shown in Table 5, our de-identification system performed much worse for PROFESSION and LOCATION on both corpora (F1-scores of them all under 90%) than other categories. For PROFESSION PHI instances, the main reason is that they appear in various formats. For example, "computer programmer" in "Works as computer programmer" and "programming" in "her programming will alleviate the pain and brings pleasure to her life" should be identified as PROFESSION instances. Our system can easily identify "computer programmer" but cannot identify "programming" as "programming" is the work a person in a profession does. For LOCATION PHI instances, a large number of instances in abbreviated forms are not identified. For example, in "...got a job at a Dunkin Donuts. Couldn't make it from school to DD...", "DD" is an abbreviation of "Dunkin Donuts", and it is not identified as an ORGANIZATION instance by our system since it is not easy to know relationship between an abbreviation and its original phrase directly. A possible solution is to design a specific module to handle abbreviations, which is one case of our future work.

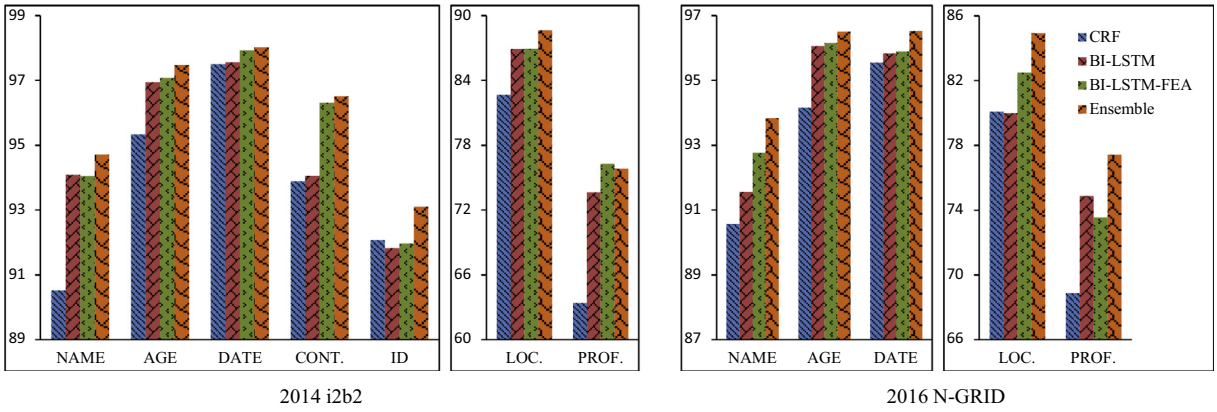

Fig. 4. The strict F1-scores of CRF, BI-LSTM, BI-LSTM-FEA and Ensemble models for each main PHI category, where CONT., LOC. and PROF. Represent the CONTACT, LOCATION and PROFESSION categories respectively. The LOCATION and PROFESSION categories are displayed in a separated subgraph as the F1-scores of them are much lower than others. The CONTACT and ID categories on the 2016 N-GRID corpus are not shown here as they are all same predicted by rules in all above methods: the F1-scores of them are 92.00% and 65.45% respectively.

In this study, we were planning to investigate effects of different word embeddings (SENNA and MEDLINE) on machine learning-based systems. However, the difference between them is very slight (the "strict" micro-averaged F1-scores of CRF-based systems using SENNA and MEDLINE on the 2016 N-GRID corpus are 88.42% and 88.37%, respectively). Therefore, we do not report the corresponding results here.

6. Conclusion

In this study, we propose a hybrid method based on RNN and CRF for de-identification of clinical notes. Experiments on the 2014 i2b2 and 2016 N-GRID corpora show that our system achieves the micro F1-scores of 96.98%, 95.11% and 98.28% under the "token", "strict" and "binary token" criteria respectively on the 2014 i2b2 test set, and 93.07%, 91.43% and 95.23% respectively on the 2016 N-GRID test set, outperforming other state-of-the-art systems and ranking first in the 2016 CEGS N-GRID NLP challenge. Among the three individual machine learning-based methods, on the whole, BI-LSTM outperforms CRF, but is inferior to BI-LSTM-FEA. However, each of them identifies certain PHI instances that cannot be identified by other two methods. Because of this, we obtain a higher overall F1-score when we combine them by an ensemble classifier. Our final system does not perform very well on PROFESSION and LOCATION PHI instances. A possible direction for further improvement is to design a module to handle abbreviations.

Acknowledgements

This paper is supported in part by grants: National 863 Program of China (2015AA015405), NSFCs (National Natural Science Foundations of China) (61573118, 61402128, 61473101, and 61472428), Special Foundation for Technology Research Program of Guangdong Province (2015B010131010), Strategic Emerging Industry Development Special Funds of Shenzhen (JCYJ201406 27163809422, 20151013161937, JSGG20151015161015297 and JCYJ20160531192358466), Innovation Fund of Harbin Institute of Technology (HIT.NSRIF.2017052), Program from the Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education (93K172016K12) and CCF-Tencent Open Research Fund (RAGR20160102). We also thank the 2016 CEGS N-GRID challenge supported by grants: NIH P50 MH106933 of Isaac Kohane and NIH 4R13LM011411 of Ozlem Uzuner.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.jbi.2017.05.023.

References

[1] A. Act, Health insurance portability and accountability act of 1996, Public Law 104 (1996) 191.
[2] Ö. Uzuner, Y. Luo, P. Szolovits, Evaluating the state-of-the-art in automatic de-identification, J. Am. Med. Inform. Assoc. 14 (5) (2007) 550–563.
[3] A. Stubbs, Ö. Uzuner, Annotating longitudinal clinical narratives for de-identification: the 2014 i2b2/UTHealth corpus, J. Biomed. Inform. 58 (2015) S20–S29.
[4] Ö. Uzuner, A. Stubbs, Practical applications for natural language processing in clinical research, J. Biomed. Inform. 58 (S) (2015) S1–S5.
[5] A. Stubbs, C. Kotfila, Ö. Uzuner, Automated systems for the de-identification of longitudinal clinical narratives: overview of 2014 i2b2/UTHealth shared task Track 1, J. Biomed. Inform. 58 (2015) S11–S19.
[6] A. Stubbs, M. Filannino, Ö. Uzuner, De-identification of psychiatric intake records: overview of 2016 CEGS N-GRID Shared Tasks Track 1, J. Biomed. Inform. 75 (2017) S4–S18.
[7] S.M. Meystre, F.J. Friedlin, B.R. South, S. Shen, M.H. Samore, Automatic de-identification of textual documents in the electronic health record: a review of recent research, BMC Med. Res. Methodol. 10 (1) (2010) 70.
[8] O. Ferrández, B.R. South, S. Shen, F.J. Friedlin, M.H. Samore, S.M. Meystre, Evaluating current automatic de-identification methods with veteran health administration clinical documents, BMC Med. Res. Methodol. 12 (1) (2012) 109.
[9] L. Deleger, K. Molnar, G. Savova, F. Xia, T. Lingren, Q. Li, K. Marsolo, A. Jegga, M. Kaiser, L. Stoutenborough, Large-scale evaluation of automated clinical note de-identification and its impact on information extraction, J. Am. Med. Inform. Assoc. 20 (1) (2013) 84–94.
[10] T. Chen, R.M. Cullen, M. Godwin, Hidden Markov model using Dirichlet process for de-identification, J. Biomed. Inform. 58 (2015) S60–S66.
[11] A. Dehghan, A. Kovacevic, G. Karystianis, J.A. Keane, G. Nenadic, Combining knowledge-and data-driven methods for de-identification of clinical narratives, J. Biomed. Inform. 58 (2015) S53–S59.
[12] Z. Liu, Y. Chen, B. Tang, X. Wang, Q. Chen, H. Li, J. Wang, Q. Deng, S. Zhu, Automatic de-identification of electronic medical records using token-level and character-level conditional random fields, J. Biomed. Inform. 58 (2015) S47–S52.
[13] H. Yang, J.M. Garibaldi, Automatic detection of protected health information from clinic narratives, J. Biomed. Inform. 58 (2015) S30–S38.
[14] G. Szarvas, R. Farkas, R. Busa-Fekete, State-of-the-art anonymization of medical records using an iterative machine learning framework, J. Am. Med. Inform. Assoc. 14 (5) (2007) 574–580.
[15] B. Wellner, M. Huyck, S. Mardis, J. Aberdeen, A. Morgan, L. Peshkin, A. Yeh, J. Hitzeman, L. Hirschman, Rapidly retargetable approaches to de-identification in medical records, J. Am. Med. Inform. Assoc. 14 (5) (2007) 564–573.
[16] X. Ma, E. Hovy, End-to-end sequence labeling via bi-directional lstm-cnns-crf, Also available at: arXiv preprint arXiv:1603.01354.
[17] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural Architectures for Named Entity Recognition, in: Proceedings of NAACL-HLT, 2016, pp. 260–270.
[18] J.P. Chiu, E. Nichols, Named entity recognition with bidirectional LSTM-CNNs, Trans. Assoc. Comput. Linguist. 4 (2016) 357–370.

[19] Z. Huang, W. Xu, K. Yu, Bidirectional LSTM-CRF models for sequence tagging, Also available at: arXiv preprint arXiv:1508.01991.
[20] L. Sweeney, Replacing personally-identifying information in medical records, the Scrub system., in: Proceedings of the AMIA Annual Fall Symposium, American Medical Informatics Association, 1996, pp. 333–337.
[21] D. Gupta, M. Saul, J. Gilbertson, Evaluation of a deidentification (De-Id) software engine to share pathology reports and clinical documents for research, Am. J. Clin. Pathol. 121 (2) (2004) 176–186.
[22] S.M. Thomas, B. Mamlin, G. Schadow, C. McDonald, A successful technique for removing names in pathology reports using an augmented search and replace method., in: Proceedings of the AMIA Symposium, American Medical Informatics Association, 2002, pp. 777–781.
[23] B.A. Beckwith, R. Mahaadevan, U.J. Balis, F. Kuo, Development and evaluation of an open source software tool for deidentification of pathology reports, BMC Med. Inform. Decis. Mak. 6 (1) (2006) 12.
[24] F.J. Friedlin, C.J. McDonald, A software tool for removing patient identifying information from clinical documents, J. Am. Med. Inform. Assoc. 15 (5) (2008) 601–610.
[25] I. Neamatullah, M.M. Douglass, H.L. Li-wei, A. Reisner, M. Villarroel, W.J. Long, P. Szolovits, G.B. Moody, R.G. Mark, G.D. Clifford, Automated de-identification of free-text medical records, BMC Med. Inform. Decis. Mak. 8 (1) (2008) 32.
[26] R. Guillen, Automated de-identification and categorization of medical records, in: i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data, Vol. 116, 2006.
[27] B. He, Y. Guan, J. Cheng, K. Cen, W. Hua, CRFs based de-identification of medical records, J. Biomed. Inform. 58 (2015) S39–S46.
[28] J.D. Lafferty, A. McCallum, F.C.N. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, in: Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001, pp. 282–289.
[29] H. Xue, S. Chen, Q. Yang, Structural support vector machine, Advances in Neural Networks-ISNN 2008 (2008) 501–511.
[30] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, B. Scholkopf, Support vector machines, IEEE Intell. Syst. Their Appl. 13 (4) (1998) 18–28.
[31] S.R. Eddy, Hidden Markov models, Curr. Opin. Struct. Biol. 6 (3) (1996) 361–365.
[32] Y. Freund, R.E. Schapire, A desicion-theoretic generalization of on-line learning and an application to boosting, in: Computational Learning Theory, Springer, 1995, pp. 23–37.
[33] B. Tang, H. Cao, Y. Wu, M. Jiang, H. Xu, Recognizing clinical entities in hospital discharge summaries using Structural Support Vector Machines with word representation features, BMC Med. Inform. Decis. Mak. 13 (Suppl 1) (2013) S1.
[34] C. Goller, A. Kuchler, Learning task-dependent distributed representations by backpropagation through structure, IEEE International Conference on Neural Networks, 1996, vol. 1, IEEE, 1996, pp. 347–352.
[35] K. Cho, B. van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: encoder decoder approaches, syntax, Semant. Struct. Stat. Transl. (2014) 103.
[36] C. dos Santos, V. Guimaraes, R.J. Niteroi, R. de Janeiro, Boosting Named entity recognition with neural character embeddings, in: Proceedings of NEWS 2015 The Fifth Named Entities Workshop, 2015, p. 25.
[37] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, J. Mach. Learn. Res. 12 (Aug) (2011) 2493–2537.
[38] X. Chen, Z. Liu, M. Sun, A Unified Model for Word Sense Representation and Disambiguation., in: EMNLP, Citeseer, 2014, pp. 1025–1035.
[39] R. Collobert, Deep Learning for Efficient Discriminative Parsing., in: AISTATS, vol. 15, 2011, pp. 224–232.
[40] D. Chen, C.D. Manning, A Fast and Accurate Dependency Parser using Neural Networks., in: EMNLP, 2014, pp. 740–750.
[41] F. Dernoncourt, J.Y. Lee, Ö. Uzuner, P. Szolovits, De-identification of patient notes with recurrent neural networks, J. Am. Med. Inform. Assoc. (2016) ocw156.
[42] K. Toutanova, D. Klein, C.D. Manning, Y. Singer, Feature-rich part-of-speech tagging with a cyclic dependency network, in: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, Association for Computational Linguistics, 2003, pp. 173–180.
[43] J.R. Finkel, T. Grenager, C. Manning, Incorporating non-local information into information extraction systems by gibbs sampling, in: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 2005, pp. 363–370.
[44] P.F. Brown, P.V. Desouza, R.L. Mercer, V.J.D. Pietra, J.C. Lai, Class-based n-gram models of natural language, Comput. Linguist. 18 (4) (1992) 467–479.
[45] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.
[46] N. Okazaki, CRFsuite: a fast implementation of conditional random fields (CRFs), 2007. URL <http://www.chokkan.org/software/crfsuite/>, 2007.
[47] R. Collobert, J. Weston, A unified architecture for natural language processing: deep neural networks with multitask learning, in: Proceedings of the 25th International Conference on Machine Learning, ACM, 2008, pp. 160–167.
[48] Y. Kim, E. Riloff, Stacked Generalization for Medical Concept Extraction from Clinical Notes, Association for Computational Linguistics, 2015.
[49] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. (TIST) 2 (3) (2011) 27.