MEDINFO 2019: Health and Wellbeing e-Networks for All
L. Ohno-Machado and B. Séroussi (Eds.)

doi:10.3233/SHTI190215

An Empirical Test of GRUs and Deep Contextualized Word Representations on De-Identification

Kahyun Lee, Michele Filannino, Özlem Uzuner

Department of Information Science and Technology, George Mason University, Fairfax, VA, United States

Abstract

De-identification aims to remove 18 categories of protected health information from electronic health records. Ideally, de-identification systems should be reliable and generalizable. Previous research has focused on improving performance but has not examined generalizability. This paper investigates both performance and generalizability. To improve current state-of-the-art performance based on long short-term memory (LSTM) units, we introduce a system that uses gated recurrent units (GRUs) and deep contextualized word representations, both of which have never been applied to de-identification. We measure performance and generalizability of each system using the 2014 i2b2/UTHealth and 2016 CEGS N-GRID de-identification datasets. We show that deep contextualized word representations improve state-of-the-art performance, while the benefit of switching LSTM units with GRUs is not significant. The generalizability of de-identification system significantly improved with deep contextualized word representations; in addition, LSTM units-based system is more generalizable than the GRUs-based system.

Keywords:

Data Anonymization, Natural Language Processing, Machine Learning

Introduction

Removing identifiable information from personal data is important for protecting privacy. The General Data Protection Regulation (GDPR) in the European Union (EU) prohibits processing of personal data without obtaining explicit consent unless identifiable information is first removed from the data. In the United States, the Health Insurance Portability and Accountability Act (HIPAA) requires removal of 18 categories of protected health information (PHI) from electronic health records (EHRs) in order to protect the privacy of patients. The task of removing such information is often referred to as de-identification. This task is often the first step that must be performed before EHRs can be shared. In this paper, we focus on the de-identification of EHRs for HIPAA compliance. This task can be accomplished both manually and automatically. However, manual de-identification is non-reusable, time-consuming, and costly. For example, Dorr et al.[1] assessed the difficulty and time costs of manual de-identification in clinical notes. On average, manual de-identification of a clinical note, with an average note length of 261 ± 352 words, required 87.3 ± 61 seconds. Douglass et al.[2] found that each human annotator could read 16,000 to 20,000 words hourly and was paid $50 per hour. Given the vast amount of EHRs and the time and financial cost of manual de-identification, fully manual de-identification is not feasible and must benefit from automated methods.

The earliest automated de-identification systems were primarily rule-based [3–5]. Researchers utilized external knowledge resources, such as lists of personal names and addresses, as well as medical term dictionaries. They also implemented regular expressions to capture morpho-lexical features of PHI.

Recently, machine-learning systems have made significant contributions to de-identification. Most machine learning models interpret de-identification as a sequence-labeling problem [6–11]. These models can handle sequences of characters, words, or sentences. For the 2014 i2b2/UTHealth shared task on de-identification [12], Yang and Garibaldi [7] earned the highest F1-score among participants using a sequence-labeling classifier based on conditional random fields (CRFs) which were complemented with customized rules. All the systems that ranked among the top five in this shared task were based on CRFs. More recently, artificial neural networks (ANNs), which are also sequence labelers, have been used for de-identification. Among various architectures of ANNs, recurrent neural networks (RNNs) have been the most common. A type of RNNs, referred to as long short-term memory (LSTM) units combined with CRFs along with manually encoded rules [11] have displayed the best performance in the 2016 CEGS N-GRID de-identification shared task [13]. Interestingly, the best systems in both shared tasks complemented their machine learning solutions with hand-crafted rules that were optimized to their datasets. Independently of these shared tasks, Dernoncourt et al. [14] built NeuroNER, and achieved state-of-the-art performance on the 2014 i2b2/UTHealth dataset. NeuroNER is also based on bi-directional LSTM (bi-LSTM) units and CRFs but includes no hand-crafted features or external resources other than GloVe word embeddings [15]. NeuroNER implements the bi-LSTM units on a character-level input to get character embeddings and concatenates character embeddings with pre-trained token embeddings. The resulting character-enhanced token embeddings are fed into the bi-LSTM units again and the sequence of probabilities are then tuned with the CRF sequence optimizer to produce the system output.

The success of NeuroNER prompts some follow up questions. The first concerns the use of different recurrent units. Although RNN with traditional neural units could handle sequence-based problems such as de-identification, they cannot effectively handle tasks that contain long-term dependencies. This is often referred to as the vanishing gradient problem. To mitigate the vanishing gradient problem, researchers have devised either enhanced learning algorithms [16] or more sophisticated activation functions [17, 18]. Among these activation functions, LSTM units are the most popular. However, in 2014, Cho et al. [18] proposed a different type of gating mechanism, namely the gated recurrent units (GRUs). Like LSTM units, GRUs can address the vanishing gradient problem, but they also have simpler structure relative to LSTM units. While LSTM units contain "forget," "update," and "output" gates, GRUs merely have "reset" and "update" gates. Despite this structural advantage and comparable performance [19], GRUs have not

been evaluated in de-identification.

The second question entails the benefit of incorporating different combinations of embeddings. Although basic word representations such as GloVe [15] and Word2Vec [20] have become the core of most recent natural language-processing (NLP) research, they face challenges when handling out-of-vocabulary (OOV) words and ambiguity, i.e., words that do not have an embedding (such as misspellings, dates, telephone numbers, and ID numbers) and words that are ambiguous (such as the word 'can' as a modal in the sentence "I can do it" and the same word as a noun in the sentence "this is a trash can"). Character embeddings could compensate for some of the weaknesses of these representations providing character-level information that can encapsulate the semantics of words. In their ablation test to investigate the contribution of each component to the performance of NeuroNER, Dernoncourt et al. [14] determined that their system performed reasonably well using only character-level embeddings (i.e., without word embeddings). However, both word embeddings and character embeddings contain limited or no information about context, which can be pivotal for capturing the true semantics of words. Recently, Peters et al. [21] devised word representations that can incorporate context into token embeddings. These "deep contextualized word embeddings" are calculated from a bi-directional language model (biLM) and are evaluated in our experiments.

Finally, despite the high performance of existing machine-learning-based de-identification systems, these systems do not generalize well to new types of data. The result of the 2016 CEGS N-GRID shared task on "sight unseen" data supports this observation. This task tested the performance of existing, unmodified de-identification systems on new types of data consisting of psychiatry records and was the first shared task of its type that utilized psychiatry data. Nine teams participated in this task. Each team could train their system on any data of their choosing. All systems were evaluated on the same previously unseen test data. The highest-performing team on this task could not reach 0.80 in terms of strict entity-based F1-score [13].

To reliably remove PHI from EHRs in practice, however, de-identification systems should generalize to new types of data that are different from what they trained on. Systems tested on the 2016 CEGS N-GRID shared task on "sight unseen" data were each trained on different sets of data, which rendered comparison of generalizability of their models impossible. Here, we present a systematic evaluation of generalizability of our systems. To the best of our knowledge, this is the first use of GRUs and deep contextualized word representations in de-identification and provides a systematic assessment of generalizability of discussed solutions.

Methods

State-of-the-art in de-identification was developed by Dernoncourt et al. [14]. This system, NeuroNER, consists of four layers, including a character-embedding layer, a token-embedding layer, a label-prediction layer, and a label-sequence-optimization layer. NeuroNER utilizes bi-LSTM units on character-level input to get character embeddings and concatenates character embeddings with pre-trained token embeddings. The resulting character-enhanced token embeddings are fed into bi-LSTM units again, and the resulting sequence of probabilities are then tuned with a CRF label sequence optimizer.

We selected NeuroNER as our baseline model. We maintained its overall pipeline, and modified its components (as described in the following experiments) to evaluate GRUs and deep contextualized word representations in de-identification, and to assess generalizability of the resulting models.

Datasets

We evaluated de-identification performance on the 2014 i2b2/UTHealth and the 2016 CEGS N-GRID de-identification data. The 2014 i2b2/UTHealth de-identification dataset (henceforth, the 2014 dataset) consists of 1,304 medical records, including discharge summaries, progress notes, doctor's notes, doctor-patient communications from 296 diabetic patients. These records contain 28,872 PHI entities in 28 types (including subtypes for some HIPAA PHI categories) [12]. The 2016 CEGS N-GRID de-identification dataset (henceforth, the 2016 dataset) includes 1,000 psychiatric intake records containing 34,364 PHI entities with the same types [13]. Table 1 summarizes the statistics of each dataset. As indicated in the table, the 2016 dataset contains more tokens and PHI per record. Considering the ratio between the average token per document and the average PHI per document, the 2014 dataset is denser than the 2016 dataset.

Both the 2014 and 2016 datasets consisted of training (60%) and test (40%) sets. We divided each of the training sets into training (40%) and validation (20%) sets. We used the training and validation sets for system development. We report results on the test sets.

Table 1 – Overview of the 2014 and 2016 Datasets

	2014 Dataset	2016 Dataset
Records	1,304	1,000
Total # of tokens	805,118	1,862,452
Average # of tokens per record	617	1,862
Total # of PHI entities	28,872	34,364
Average # of PHI per record	22	34

Evaluation Metrics

For evaluation, we used micro-averaged precision, recall, and F1-score computed using the following equations:

- Precision (P) = true positives / (true positives + false positives)
- Recall (R) = true positives / (true positives + false negatives)
- F1-score (F1) = 2*P*R / (P+R)

Organizers of the 2014 and 2016 de-identification shared tasks measured multiple versions of these metrics (strict/relaxed/overlap with token-based/entity-based matching) over two sets of PHI types (i2b2 PHI vs. HIPAA PHI). Strict, relaxed, and overlap measurements differ in terms of their acceptance of exact or inexact textual spans corresponding to PHI. Token-based versus entity-based evaluation refers to whether the complete string or individual words in the string are considered for de-identification. i2b2 PHI and HIPAA PHI differ in terms of granularity of PHI. i2b2 categories classify the 18 HIPAA categories into sub-types (e.g., a location HIPAA PHI type can be classified into street, city, state, zip i2b2 types). In this paper, we evaluate performance with the strictest settings—strict entity-based matching over i2b2 PHI.

System Structure

NeuroNER serves as both the starting point and the benchmark for the experiments presented in this paper. Our goal is to improve performance of this state-of-the-art system on de-identification and to assess its generalizability under different conditions. We experiment with different RNNs and

embeddings in order to evaluate the contribution of each to system performance and generalizability.

Our system accepts tokenized narrative text as input. For this, the narratives are first pre-processed using spaCy [22] for sentence detection and tokenization. Given the pre-processed narratives and NeuroNER, we experimented with bi-LSTM units and bi-GRUs. LSTM units and GRUs are alike in that they both incorporate history into their predictions. But as previously mentioned, GRUs have simpler structure than LSTM units. As a result, GRUs have fewer parameters than LSTM units, and can converge faster [19]. In terms of their performance, the superiority or inferiority of either LSTM units or GRUs remains a matter of debate [19, 23], while the merits of both units relative to the traditional RNNs units (for example, tanh) are undeniable [24,25].

In order to evaluate the performance of these units, we tested three different types of embeddings with them. Pre-trained word embeddings [12,17] are widely used and have become a core element in recent machine-learning-based NLP research; however, they can fall short on OOV and ambiguous words. One way of resolving these cases would entail the use of character-enhanced token embeddings or deriving from deep contextualized word representations. Rather than training a word vector for each token, deep contextualized word representations train a context function that is based on two-layer biLMs with character convolutions and that returns a flexible word vector [21]. This function assigns a different vector to each word based on its morphological features and context; even the same two words in the same sentence can have different word vectors. Deep contextualized word representations can mitigate both OOV and ambiguous word problems. We used 1,024-dimensional deep contextualized word representations in addition to character embeddings and word embeddings. We processed our datasets (using only the training and validation portions) with a pre-trained model (which is a context function trained on the 1B Word Benchmark [26]) to extract deep contextualized word representations.

To investigate the contribution of different embeddings to different RNNs on de-identification, we experimented with all possible combinations of items within the four layers of NeuroNER architecture. The layers are summarized in Table 2.

Table 2 – Experiment Items in Each Architectural Layer of NeuroNER

Location	Experiment items
4th layer	CRF label sequence optimizer
3rd layer	bi-LSTM units vs. bi-GRUs
2nd layer	word embeddings (WEs) and/or deep contextualized word representations (DEs)
1st layer	character embeddings (CEs)

Training and Hyperparameters

We tuned our hyperparameters using the training set of each dataset. NeuroNER accomplished its optimal performance on both datasets with these parameters:

- character-embedding dimension: 25
- character-based token-embedding dimension: 25
- token-embedding dimension: 100
- label-prediction dimension: 100
- dropout probability: 0.5

In addition to the above hyperparameters, the dimension of the deep contextualized word representations was 1,024. Therefore, the inputs to the label-prediction layer entail a minimum of 25 dimensions (in the case that only character embeddings are tested) and a maximum of 1,149 dimensions (in the case that all three embeddings are tested). When training, iteration was stopped if there was no improvement in the F1-score on the validation set for 10 epochs.

Results

Performances of both LSTM units-based and GRUs-based RNNs are presented in Tables 3 and 4. The statistical significance of F1-score difference from the baseline model was tested using approximate randomization [27] with 9,999 shuffles. Statistically significant F1-score differences at the level of 0.01 are asterisked.

Table 3 indicates the performance of LSTM units-based RNNs with different combinations of embeddings. The baseline performance of NeuroNER is underlined, and the precision, recall, and F1-score of the highest-performing combination for each dataset is in bold font. In the 2014 dataset, the role of character embeddings was pivotal. Without word embeddings, LSTM units-based RNNs accomplished 89.70 for F1-score. Any combinations that included deep contextualized word representations significantly outperformed the combination of character embeddings and word embeddings, which were our baseline, in F1-score. Even the system that used deep contextualized word representations alone could outperform this baseline in F1-score. The highest F1-score achieved with LSTM units-based RNNs was 92.82, which is 1.15 higher than the baseline. In the 2016 dataset, the system with only character embeddings underperformed relative to the system with only word embeddings. Deep contextualized word representations worked effectively on the 2016 dataset as well, but it did not surpass the baseline system until it was combined with the other two embeddings. The highest F1-score was 89.00. In both the 2014 and 2016 datasets, the optimal performances were observed when all three embeddings were combined.

Table 3 – Performance of LSTM Units-Based RNNs

Embeddings	2014 Dataset			2016 Dataset		
	P	R	F1	P	R	F1
CEs	92.83	86.77	89.70*	84.96	78.58	81.65*
WEs	85.46	79.09	82.15*	85.92	79.99	82.85*
DEs	93.50	90.79	92.13*	88.23	86.56	87.39*
CEs+WEs	92.39	90.97	91.67	89.15	87.09	88.11
CEs+DEs	93.14	90.84	91.98*	88.65	86.95	87.79
WEs+DEs	93.34	90.91	92.11*	87.99	87.92	87.95
All three	94.33	91.36	92.82*	90.88	87.20	89.00*

Table 4 – Performance of GRUs-Based RNNs

Embeddings	2014 Dataset			2016 Dataset		
	P	R	F1	P	R	F1
CEs	91.98	87.30	89.58*	84.77	77.20	80.81*
WEs	86.86	77.81	82.09*	84.66	78.73	81.59*
DEs	93.04	91.23	92.13*	88.48	85.99	87.22*
CEs+WEs	93.77	90.49	92.10*	89.66	86.39	87.99*
CEs+DEs	93.63	90.33	91.95*	90.57	86.58	88.53*
WEs+DEs	94.07	90.74	92.37*	89.97	87.51	88.72*
All three	93.59	91.65	92.61*	89.11	87.10	88.09

Table 4 indicates the performance of the GRUs-based RNNs. Here, character embeddings were pivotal in the 2014 dataset but not on the 2016 dataset. The combination of GRUs with deep contextualized word representations achieved a good performance. The optimal F1-score in the 2014 dataset was received when all three embeddings were combined, while the combination of word embeddings and deep contextualized

word representations exhibited the second highest performance. The differences in F1-score are statistically significant at P=.008. Regarding the 2016 dataset, the optimal performance was produced when word embeddings and deep contextualized word representations were combined; combining character embeddings with these two embeddings deteriorated the performance of the system.

Comparing Tables 3 and 4, we see that GRUs demonstrated a comparable performance relative to the LSTM units and even outperformed the LSTM units in some combinations. Notably, the F1-score increased by 0.43 merely as a result of substituting the LSTM units with the GRUs for the 2014 dataset (which does not occur in the 2016 dataset). The differences in F1-score are statistically significant at P=0.002. However, the highest performance with the GRUs-based RNNs could not surpass that of the LSTM units-based RNNs for both datasets. The differences in F1-scores are statistically significant at P=0.004.

We also investigated the generalizability of both LSTM units-based and GRUs-based RNNs systems. To test the generalizability of each combinatio, we evaluated the performance of the model trained using the 2014 training set on the 2016 test set.

Table 5 – Generalizability of LSTM Units-Based RNNs

Embeddings	2014→2016			2016→2014		
	P	R	F1	P	R	F1
CEs	67.25	55.17	60.61*	57.67	62.76	60.11*
WEs	57.57	59.81	58.67*	57.08	42.65	48.82*
DEs	74.56	69.97	72.19*	70.95	69.51	70.22*
CEs+WEs	70.62	66.01	68.24	68.62	67.09	67.85
CEs+DEs	75.95	70.96	73.37*	69.37	70.84	70.10*
WEs+DEs	74.97	70.79	72.82*	70.92	70.42	70.67*
All three	77.51	68.30	72.61*	71.98	70.90	71.44*

Table 6 – Generalizability of GRUs-Based RNNs

Embeddings	2014→2016			2016→2014		
	P	R	F1	P	R	F1
CEs	72.85	54.02	62.04*	64.24	63.46	63.85*
WEs	61.08	58.87	59.95*	50.62	44.36	47.28*
DEs	71.78	70.17	70.97*	68.76	66.11	67.40*
CEs+WEs	76.68	66.82	71.41*	66.95	66.20	66.57*
CEs+DEs	74.55	71.04	72.75*	70.81	69.09	69.94*
WEs+DEs	76.82	69.19	72.81*	69.58	69.37	69.47*
All three	72.75	71.72	71.93*	73.18	69.12	71.09*

As Tables 5 and 6 demonstrate, combinations that performed effectively on training data also work effectively on new types of data. In addition, the introduction of deep contextualized word representations significantly increased the generalizability of both LSTM units and GRUs-based systems on both datasets. The differences in F1-score of the baseline and the models that include deep contextualized word representations are statistically significant at P<0.001. The gain in generalizability over the baseline is more significant than the gain in performance. The performance increase from the baseline to the highest F1-score of LSTM units-based system within the 2014 dataset was 1.15, while the increase in the performance while going across datasets was 5.13.

Discussion

Error Analysis

The distribution of PHI found by each RNN is shown in Table 7. Overall, both RNNs succeeded in finding 9,871 entities from the 2014 test set and 9,950 entities from the 2016 test set. In contrast, both units missed 1,302 entities in the 2014 test set and 3,339 entities in the 2016 test set. To investigate the strengths and weaknesses of each RNN, the number of entities found by only one of the units were counted. For example, "only GRUs found" indicates the number of PHI entities that all embeddings combinations with LSTM units failed to detect, but all embeddings combinations with the GRUs could find. These entities can be only found by GRUs regardless of embedding types, and they may reveal the empirical difference between LSTM units and GRUs.

Table 7 – Distribution of PHI by RNNs

	2014 Dataset	2016 Dataset
Total PHI in the test set	11,462	13,519
Both found	9,871	9,950
Both failed	1,302	3,339
Only GRUs found	243	104
Only LSTM units found	46	126

We extracted the sentences that include "only LSTM units found" and "only GRUs found" PHI entities and manually reviewed their characteristics. In many "only GRUs found" PHIs, the entities were directly adjacent to special characters such as slashes (/), colons (:), commas (,), and hyphens (-). For example, 102 DOCTOR entities, which are typically in the form of 'initial/name1/name2' or 'initial:name', could be found only by GRUs-based systems. It appears that GRUs are less sensitive to directly adjacent tokens in predicting the label of the current token. In contrast, PHI only found by LSTM units contain more contextually complex sentences. For example, in 41 cases, seasons and years (both are in DATE category) in sentences such as "healthy until spring of 0314 2126," "in the spring of 2128," and "in the summer of 2096" could not be found by GRUs, but they were detected by LSTM units. However, these findings need to be taken with a grain of salt given the small sample size.

Table 8 – Distribution of PHI by Embeddings

	2014 Dataset	2016 Dataset
Total PHI in the test set	11,462	13,519
Found only without CEs	347	175
Found only with CEs	131	249

Table 8 indicates the distribution of PHI for deep contextualized word representations with both RNNs. A total of 380 PHI entities were additionally found with these embeddings. These PHI entities were primarily centered around several entity types, such as PROFESSION (19%), HOSPITAL (15%), ORGANIZATION (15%), PATIENT (12%), and DOCTOR (11%). This distribution differs from the original PHI distribution of the test set: PROFESSION (5%), HOSPITAL (9%), ORGANIZATION (3%), PATIENT (7%), and DOCTOR (14%). From a practical perspective, 'PATIENT-DOCTOR' and 'HOSPITAL-ORGANIZATION' pairs are difficult to be differentiated from each other because of the semantic similarity between these PHI categories. Through manual review, we found these PHI entities were in complete sentences in most cases. This illustrates that a pre-trained context function returns better word representations when inputs are lengthy and sufficiently informative to make a machine understand context. Gains in the 2016 dataset are nearly twice as abundant as those in the 2014 dataset perhaps because EHRs in the 2016 dataset are more organized and structured than those in the 2014 dataset. As a result, the sentence detector worked more effectively on the 2016 dataset, providing cleaner input to the RNNs.

Miscellaneous

On average, GRUs show higher per-category F1-score on entity types CITY, COUNTRY, DOCTOR, HOSPITAL, IDNUM, PHONE, PROFESSION, and STATE whereas LSTM units perform better on AGE, DATE, MEDICALRECORD, ORGANIZATION, PATIENT, STREET, USERNAME, and ZIP. Interestingly, deep contextualized embeddings achieve higher per-category F1-score than both character and word embeddings regardless of entity types. Theoretically, the convergence of GRUs-based systems should be faster than LSTM units-based systems because GRUs have simpler structure and fewer parameters than LSTM units. This was confirmed by the experiments. For the 2014 dataset, convergence required an average of 37 epochs for the LSTM units-based systems, whereas for GRUs-based systems, convergence required an average of 28 epochs. Similarly, in the 2016 dataset, LSTM units required an average of 32 epochs, whereas GRUs required 30 epochs.

Conclusions

To automate de-identification tasks, de-identification must be highly accurate and generalizable. State-of-the-art de-identification systems are based on RNNs with bi-LSTM units. This study investigated avenues for improving the state-of-the-art by modifying types of recurrent units and embeddings. It also investigated how these modifications affect generalizability. We found that substituting LSTM units with the GRUs cannot significantly improve performance or generalizability. However, this does not imply the inferiority of GRUs relative to LSTM units on de-identification. GRUs outperformed LSTM units in some PHI types and could complement LSTM units in an ensemble. They could also converge to optimal parameters faster than LSTM units. The introduction of deep contextualized word representations on top of character embeddings and word embeddings was certainly helpful in increasing both the state-of-the-art performance and generalizability.

References

[1] D.A. Dorr, W.F. Phillips, S. Phansalkar, S.A. Sims, and J.F. Hurdle, Assessing the difficulty and time cost of de-identification in clinical narratives, *Methods Inf. Med.* **45** (2006) 246–252.

[2] M. Douglass, G.D. Clifford, a Reisner, G.B. Moody, and R.G. Mark, Computer-assisted De-identification of free text in the MIMIC II database, Comput. Cardiol. 31 (2004) 341–344.

[3] L. Sweeney, Replacing personally-identifying information in medical records, the Scrub system., *AMIA Annu Symp Proc.* (1996) 333–7.

[4] S.M. Thomas, B. Mamlin, G. Schadow, and C. McDonald, A successful technique for removing names in pathology reports using an augmented search and replace method., *AMIA Annu Symp*. (2002) 777–81.

[5] F.J. Friedlin, and C.J. McDonald, A Software Tool for Removing Patient Identifying Information from Clinical Documents, *J. Am. Med. Informatics Assoc.* **15** (2008) 601–610.

[6] O. Ferrández, B.R. South, S. Shen, F. Jeffrey Friedlin, M.H. Samore, and S.M. Meystre, BoB, a best-of-breed automated text de-identification system for VHA clinical documents, J. Am. Med. Informatics Assoc. **20** (2013) 77–83. doi:10.1136/amiajnl-2012-001020.

[7] H. Yang, and J.M. Garibaldi, Automatic detection of protected health information from clinic narratives, *J. Biomed. Inform.* **58** (2015) S30–S38.

[8] B. He, Y. Guan, J. Cheng, K. Cen, and W. Hua, CRFs based de-identification of medical records, *J. Biomed. Inform.* **58** (2015) S39–S46.

[9] Z. Liu, Y. Chen, B. Tang, X. Wang, Q. Chen, H. Li, J. Wang, Q. Deng, and 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.

[10] H.-J. Lee, Y. Wu, Y. Zhang, J. Xu, H. Xu, and K. Roberts, A hybrid approach to automatic de-identification of psychiatric notes, *J. Biomed. Inform.* (2017).

[11] Z. Liu, B. Tang, X. Wang, and Q. Chen, De-identification of clinical notes via recurrent neural network and conditional random field, *J. Biomed. Inform.* **75** (2017) S34–S42.

[12] A. Stubbs, C. Kotfila, and 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.

[13] A. Stubbs, M. Filannino, and 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.

[14] F. Dernoncourt, J.Y. Lee, O. Uzuner, and P. Szolovits, De-identification of patient notes with recurrent neural networks, *J. Am. Med. Informatics Assoc.* **24** (2017) 596–606.

[15] J. Pennington, R. Socher, and C. Manning, Glove: Global Vectors for Word Representation, *Proc. 2014 Conf. Empir. Methods Nat. Lang. Process.* (2014) 1532–1543.

[16] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, Advances in optimizing recurrent networks, *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.* (2013) 8624–8628.

[17] S. Hochreiter, and J. Schmidhuber, Long Short-Term Memory, *Neural Comput.* **9** (1997) 1735–1780.

[18] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, (2014).

[19] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, (2014) 1–9.

[20] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, Distributed Representations of Words and Phrases and their Compositionality, (2013) 1–9.

[21] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, Deep contextualized word representations, (2018).

[22] M. Honnibal, and I. Montani, spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing, To Appear. (2017).

[23] D. Bahdanau, K. Cho, and Y. Bengio, Neural Machine Translation by Jointly Learning to Align and Translate, (2014) 1–15.

[24] R. Jozefowicz, W. Zaremba, and I. Sutskever, An Empirical Exploration of Recurrent Network Architectures, in Proc. 32nd Int. Conf. Mach. Learn., Lille, France, 2015.

[25] A. Karpathy, J. Johnson, and L. Fei-Fei, Visualizing and Understanding Recurrent Networks, (2015) 1–12.

[26] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, T. Robinson, P. Koehn, P. Koehn, and T. Robinson, One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling, INTERSPEECH. (2014).

[27] E.W. Noreen, Computer-intensive Methods for Testing Hypotheses: An Introduction, John Wiley & Sons, Inc, New York, 1989.

Address for correspondence

Kahyun Lee

klee70@gmu.edu