Drug Safety (2019) 42:123–133
https://doi.org/10.1007/s40264-018-0761-0

ORIGINAL RESEARCH ARTICLE

MADEx: A System for Detecting Medications, Adverse Drug Events, and Their Relations from Clinical Notes

Xi Yang[1] · Jiang Bian[1] · Yan Gong[2] · William R. Hogan[1] · Yonghui Wu[1]

Published online: 2 January 2019

Abstract

Introduction Early detection of adverse drug events (ADEs) from electronic health records is an important, challenging task to support pharmacovigilance and drug safety surveillance. A well-known challenge to use clinical text for detection of ADEs is that much of the detailed information is documented in a narrative manner. Clinical natural language processing (NLP) is the key technology to extract information from unstructured clinical text.

Objective We present a machine learning-based clinical NLP system—MADEx—for detecting medications, ADEs, and their relations from clinical notes.

Methods We developed a recurrent neural network (RNN) model using a long short-term memory (LSTM) strategy for clinical name entity recognition (NER) and compared it with baseline conditional random fields (CRFs). We also developed a modified training strategy for the RNN, which outperformed the widely used early stop strategy. For relation extraction, we compared support vector machines (SVMs) and random forests on single-sentence relations and cross-sentence relations. In addition, we developed an integrated pipeline to extract entities and relations together by combining RNNs and SVMs.

Results MADEx achieved the top-three best performances (F1 score of 0.8233) for clinical NER in the 2018 Medication and Adverse Drug Events (MADE1.0) challenge. The post-challenge evaluation showed that the relation extraction module and integrated pipeline (identify entity and relation together) of MADEx are comparable with the best systems developed in this challenge.

Conclusion This study demonstrated the efficiency of deep learning methods for automatic extraction of medications, ADEs, and their relations from clinical text to support pharmacovigilance and drug safety surveillance.

Key Points

Combining recurrent neural networks and support vector machines in a hybrid system achieved good performance in detecting medications, adverse drug events, and their relations from clinical notes.

Deep learning models are able to learn high-level feature representations without human intervention.

When no validation data are provided, having more samples in training may be more important than finding a local maximum.

Part of a theme issue on "NLP Challenge for Detecting Medicationand Adverse Drug Events from Electronic Health Records (MADE1.0)" guest edited by Feifan Liu, Abhyuday Jagannatha and Hong Yu.

✉ Yonghui Wu
yonghui.wu@ufl.edu

[1] Department of Health Outcomes and Biomedical Informatics, College of Medicine, University of Florida, Gainesville, FL, USA

[2] Department of Pharmacotherapy and Translational Research and Center for Pharmacogenomics, College of Pharmacy, University of Florida, Gainesville, FL, USA

1 Introduction

Adverse drug events (ADEs) are defined as injuries caused by medical intervention related to drugs [1], and are increasingly common in the US and around the world [2, 3]. A recent study examined the incidence rate of ADEs

using a US nationwide inpatient sample dataset from 2008 to 2011 and reported an average incidence rate of 6.28%, with an increasing trend from 5.97% (at 2008) to 6.82% (at 2011) [4]. Similar studies reported an incidence rate of approximately 6–8% in Saudi Arabia, 3.22% in England, and 4.78% in Germany [3, 5, 6]. ADEs are reported to increase the healthcare cost, length of stay, and in-hospital mortality rates [2]; however, a large majority of ADEs are preventable. Early detection and prevention of ADEs is expected to result in safer and higher quality healthcare, reduce healthcare cost, and improve healthcare outcome [4]. Narrative clinical text contains detailed treatment and response information from clinical practice and could be used to detect ADEs for pharmacovigilance and drug safety surveillance.

Unstructured clinical text has been increasingly used for clinical and translational research as it contains detailed patient information that cannot be captured in abstracted medical codes [7, 8]. A well-known challenge to use unstructured clinical text is that much of the detailed information is documented in a narrative manner, which is not directly accessible. Clinical natural language processing (NLP) is the key technology to extract information from unstructured clinical text to support various clinical studies and applications that depend on structured data. To use narrative clinical text for the detection of ADEs, the clinical NLP systems need to (1) identify the mentions of medications, ADEs, and their attributes—a typical clinical name entity recognition (NER) [9] task; and (2) determine their relations (e.g. which medication induced the ADE)—a relation extraction [10] task. The clinical NLP community has organized open challenges such as i2b2 (The Center for Informatics for Integrating Biology and the Bedside) challenges [11, 12], SemEval (International Workshop on Semantic Evaluation) challenges [13], and ShARe/CLEF eHealth challenges [14], to examine current NLP methods for clinical NER and relation extraction. Most NLP systems approach the clinical NER and relation extraction using machine learning-based methods. Researchers have applied various machine learning methods such as conditional random fields (CRFs) [15], support vector machines (SVMs) [16], structured SVMs (SSVMs) [17], and hybrid methods.

Clinical NER is a fundamental task to extract clinical concepts of interest (e.g. medications, ADEs) from clinical narratives [9]. Researchers have developed various NER algorithms and applied them in general clinical NLP systems. Early clinical NLP systems such as MetaMap [18], MedLEE [19], and KnowledgeMap [20] applied rule-based methods that rely on existing medical vocabularies. Later, many researchers explored machine learning models and reported improved performance. Machine learning models approach clinical NER as a sequence labeling problem—finding the best label sequence (e.g. BIO tag sequence: B—the beginning of a concept, I—words inside a concept, O—words outside a concept) for a given input sequence (words from clinical text). The machine learning algorithms sequentially scan each of the input words and determine the best label sequence according to the context features from surrounding words. Most top-performing clinical NER methods are based on machine learning models, where CRFs and SSVMs are among the most popular solutions. For example, de Bruijn et al. [21] developed the best-performing clinical NER system using a semi-Markov Hidden Markov model (HMM) in the 2010 i2b2 challenge: task 1—a concept extraction task focused on the extraction of problems, treatments and laboratory tests; Zhang et al. [22] developed the best-performing NER system using an ensemble model of CRFs and SSVMs in the 2014 SemEval open challenge: task 7—analysis of clinical text; and Tang et al. [23] contributed the best-performing NER system using SSVMs in the 2013 ShARe/CLEF eHealth Challenges on detection of disorder names from clinical notes. Recently, deep learning methods are emerging as new state-of-the-art solutions for clinical NER. Deep learning-based NER methods applied deep network architectures to learn multiple levels of data representation, which is different from the traditional machine learning methods where features were manually designed by researchers. Researchers have explored deep learning models for information extraction from biomedical literature and narrative clinical notes. For biomedical literature, Le et al. [24] reported an improved performance using a CRF-biLSTM neural network for NER. Habibi et al. [25] also applied a similar Long Short-Term Memory–Conditional Random Fields (LSTM-CRF) model and reported good performance. For narrative clinical text, Liu et al. [26] and Jagannatha et al. [27] examined recurrent neural networks (RNNs) for clinical NER. We have also examined convolutional neural networks (CNNs) [28] and RNNs [29] in our previous studies.

Relation extraction is a critical NLP task to understand the semantic relations between clinical concepts [10]. Compared with clinical NER, the feature selection for relation extraction is not that straightforward. Another critical challenge of relation extraction is that the searching space is very large—relation extraction systems have to consider the combinations among all clinical concepts in a document. Researchers have applied both supervised machine learning methods such as SVMs [16], kernel methods [30, 31], and tree kernel methods [32], and semi-supervised machine learning methods such as Dual Iterative Pattern Relation Expansion (DIPRE) [33] for relation extraction. Most state-of-the-art relation extraction methods in the medical domain are based on machine learning models. For example, de Bruijn et al. [21] developed the best-performing relation

extraction system in the 2010 i2b2 relation challenge using a maximum entropy model; Tang et al. [34] developed the best-performing temporal relation extraction system using a hybrid SVM model in the 2012 i2b2 challenge on temporal relations; and Xu et al. [35] developed the best-performing system using a customized CRFs model in the BioCreative V chemical-induced disease relation challenge. Recent studies from the general NLP domain reported that deep learning models, especially CNN models based on word embeddings and positional embeddings, outperformed traditional machine learning methods on relation extraction [10].

Previous clinical NLP challenges have designed NER tasks to extract clinical concepts such as problems, treatments, and laboratory tests from clinical text. For relation extraction, the i2b2 2010 challenge [11] examined the extraction of treatment relation, test relation (test conducted to investigate medical problems), and medical problem relation (medical problems that describe or reveal aspects of the same medical problem, e.g. *Azotemia* presumed secondary to *sepsis*); the i2b2 2012 challenge [12] examined the temporal relation (how medical events related to each other in the clinical timeline, e.g. before, after). In 2015, the BioCreative V open challenge [36] organized a relation extraction task to extract the chemical–disease relation (CDR) from biomedical literature. In 2018, the University of Massachusetts Medical School organized an NLP challenge for detecting Medication and Adverse Drug Events from electronic health records (MADE1.0). The MADE1.0 challenge has three subtasks: (1) a clinical NER task to extract medications, ADEs, and their attributes; (2) a relation extraction task to extract relations among the detected clinical concepts; and (3) an integrated task that combines subtasks (1) and (2). To the best of our knowledge, this is the first open challenge on extracting medications, ADEs, and their relations from a large clinical corpus. The CDR task of BioCreative V is related to this challenge but it focused on the chemical-induced diseases from the biomedical literature. In this article, we present the Medication and Adverse Drug Events Extraction system (MADEx) developed for the MADE1.0 challenge. MADEx consists of two modules: (1) a clinical NER module to recognize medication names and attributes (dosage, frequency, route, duration), as well as ADEs, indications, and other signs and symptoms; (2) a relation extraction module to identify relations between medications and attributes, as well as relations between medications and ADEs, indications, and other signs and symptoms. MADEx achieved a top-three best performance for the NER task using a deep learning method, demonstrating the effectiveness of deep learning approaches.

Table 1 Overall statistics of the datasets

Dataset	Notes	Entities	Relations
Training	876	67,781	23,047
Test	213	11,333	4128

Table 2 Distribution of relations in the training and test sets

Relation	Entity 1	Entity 2	Counts	
			Training	Testing
Adverse	Drug	ADE	2055	511
do	Drug	Dose	5150	863
du	Drug	Duration	901	146
fr	Drug	Frequency	4407	728
Manner/route	Drug	Route	2544	454
Reason	Drug	Indication	4530	871
Severity_type	SSLIF	Severity	2909	390
Severity_type	ADE	Severity	282	37
Severity_type	Indication	Severity	269	128

ADE adverse drug event, *do* dose relation between a drug and its dose, *du* duration, *fr* frequency, *SSLIF* other signs, symptoms, and diseases that are not an *ADE* or an indication

2 Methods

2.1 Dataset

The MADE1.0 organizers developed a corpus for clinical NER and relations extraction using a total of 1089 de-identified clinical notes. Annotators labeled medications and their attributes, ADEs, indications, and other signs and symptoms, and the relations among them. A total of 79,114 entities and 27,175 relations were annotated and represented using the *BioC* format [37]. The relations were annotated at the document level that may cross multiple sentences. The corpus was divided into a training set of 876 notes and a test set of 213 notes. Table 1 shows the overall statistics, while Table 2 provides detailed distribution of relations among all relation types for the training and test sets.

2.2 The MADEx System

The MADEx system applied machine learning methods to extract clinical concepts and their relations. We developed a recurrent neural network (RNN)-based clinical NER module using the long short-term memory (LSTM) strategy [38] with a CRFs layer—the LSTM-CRFs model [39]. We also implemented standard deep learning techniques, including bi-directional LSTM, character-level embedding, and dropout. Using different training strategies, we developed two

LSTM-CRFs models and compared them with a CRF—another widely used machine learning method for NER. For the relation extraction module, we first developed heuristic rules to generate candidate pairs from the detected clinical concepts and then applied a hybrid SVM model to determine whether there was a relation between the entities and to classify the relation types. We compared two widely used machine learning models for relation classification—SVMs and random forests (RFs). For the integrated task, we developed an NLP pipeline to integrate the two modules into a unified system. Details are described in the following sections.

2.2.1 Name Entity Recognition (NER) Module

2.3 Workflow of the NER Module

Figure 1 shows the workflow of the NER module. The NER module consists of a pre-processing pipeline, a machine learning-based clinical NER, and a post-processing pipeline. The pre-processing pipeline performed sentence boundary detection and tokenization to normalize the raw clinical notes. Since the sentence boundary detection and tokenization will change the offsets (the start and end position of clinical concepts in the clinical text), the NER module tracked all the entities offset using position mapping files. As the training data were provided using the *BioC* format [37], but the machine learning requires *BIO* format, we developed a pipeline to convert the annotation from *BioC* to *BIO*. The NER algorithm scanned the normalized notes and detected clinical concepts using pre-trained machine learning models. The post-processing pipeline mapped the detected clinical concept back to its original position, converted the predictions to *BioC* format, and dumped the results to XML files.

2.4 Machine Learning-Based NER Methods

We applied a state-of-the-art deep learning-based NER method, the LSTM-CRFs [39], and compared it with another widely used machine learning method, CRFs.

2.5 Long Short-Term Memory–Conditional Random Fields (LSTM-CRFs) Model

The LSTM-CRFs model is a special implementation of RNNs designed for sequential data composed of consecutive vectors. Different from other feed-forward neural networks, RNNs have loops in their network architectures, which enable RNNs to utilize the long-distance dependencies from previous information. Until now, the RNNs implemented using the LSTM strategy is reported to be the state-of-the-art method for NER. The LSTM implementation designed several computational functions to control the mixing of previous information with current information. In this study, we adopted an LSTM-CRFs architecture from Lample et al. [39] with the following implementation:

$$i_t = \text{sigmoid}\left(W_{xi}x_t + W_{\boldsymbol{hi}}\boldsymbol{h}_{t-1} + W_{ci}c_{t-1} + b_i\right), \quad (1)$$

$$c_t = (1 - i_t) \odot c_{t-1} + i_{t\odot\tanh}\left(W_{xc}x_t + W_{\boldsymbol{hc}}x_{t-1} + b_c\right), \quad (2)$$

$$o_t = \text{sigmoid}\left(W_{xo}x_t + W_{\boldsymbol{ho}}\boldsymbol{h}_{t-1} + W_{co}c_t + b_o\right), \quad (3)$$

$$\boldsymbol{h}_t = o_{t\odot\tanh(c_t)}. \quad (4)$$

The LSTM-CRFs model has a character-level embedding layer, a character-level bi-directional LSTM layer, a word embedding layer, a word-level bi-directional LSTM layer, and a CRFs layer for sequence labeling. To handle unknown words, we collected the low-frequency words (words that appeared only once in the training) and dynamically assigned them as 'unknown' according to a probability of 0.5 during training. More specifically, before feeding a sentence for training, we randomly generated a probability between 0 and 1 for each of the low-frequency words. According to

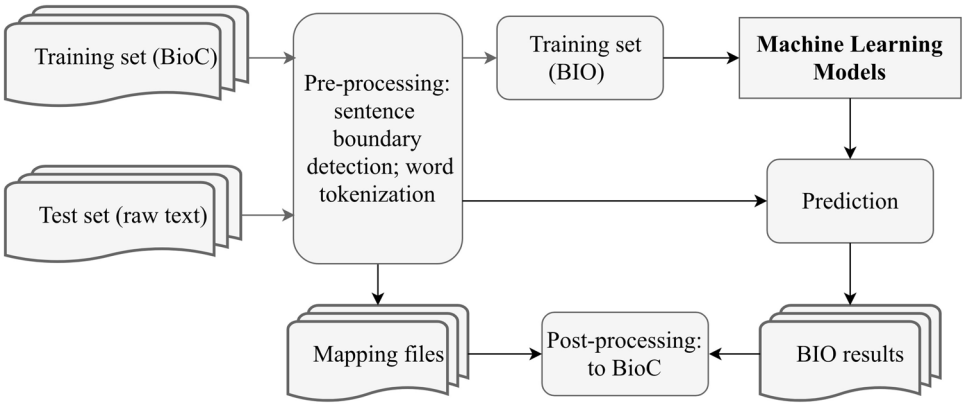

Fig. 1 Workflow of the NER module

the probability, we either replaced the low-frequency word as 'unknown' (probability > 0.5) or kept it unchanged (probability ≤ 0.5). Thus, we were able to train the embedding for 'unknown'. During prediction, we replaced all the words that were not covered by the training corpus as 'unknown'. For the training of LSTM layers, we did not use the sentence start and end paddings, but we did consider the transition probability from the sentence start to a word, and the transition probability from the last word to the sentence end, in the CRFs layer. The training strategy of our system is different from the typical LSTM-CRFs. The typical training procedure split part of the training as a validation and selected the best model according to the performance on the pre-split validation set. We developed a two-stage training procedure, including stage 1 to optimize the model parameters according to the performance on the pre-split validation set; and stage 2 to merge the pre-split validation set back to the training set and retrain a new model using the parameters and stop iterations optimized by stage 1. The typical training of LSTM-CRFs was able to find a local maximum, but it had less training samples as part of the training samples were split for validation. Our training strategy kept more samples in training (as we merged the validation back to training at stage 2), but the final model may not be at a local maximum. Our assumption is that keeping more samples in training may be more important than finding a local maximum.

2.6 CRFs Model

CRF is another popular machine learning model for clinical NER as it is intrinsically designed for sequence labeling problems by modeling the relationships between neighbor tokens in the sequence. In this study, we utilized the CRFs algorithm implemented in the CRFsuite library (http://www.chokkan.org/software/crfsuite/). We used machine learning features that were reported to be useful for clinical NER in previous studies, including word n-grams, prefixes, suffixes, word shape (combination patterns of uppercase and lowercase letters, numbers), sentence-level features (sentence length, whether the sentence is a part of a list), brown clustering, and discrete word embedding [40, 41]. The discrete word embedding features were derived by converting the real numbers in the word embedding into discrete categories in [POSITIVE, NEGATIVE, NEUTRAL]. For each dimension of word embedding, we calculated the positive mean value, i.e. the arithmetic mean among all positive values of this dimension, and the negative mean, i.e. the arithmetic mean among all negative values of this dimension. For each value in this dimension, we compared it with the positive mean and negative mean. If the value is bigger than the positive mean, we replaced it as 'POSITIVE'; if less than the negative mean, we replaced it as 'NEGATIVE'. The values between the negative and positive means were replaced as 'NEUTRAL'. We trained CRF models using all 876 notes in the training set, and optimized the parameters using fivefold cross-validation.

2.6.1 Relation Extraction Module

Given clinical concepts, the goal of the relation extraction module is to determine whether there is a relation among the concepts and to identify the relation types. We approached relation extraction as a classification task, with each relation type as an individual class. In this study, there are a total of eight classes, including seven classes shown in Table 2 and a 'non-relation' class denoting there is no relation between entities. The relation extraction module consists of three parts: a pre-processing pipeline, a classifier, and a post-processing pipeline. Figure 2 shows the workflow of the relation extraction module. The pre-processing pipeline generated candidate pairs of clinical concepts using heuristic rules based on the permutation among all concepts. The classifier then assigned one of the eight classes for each candidate pair. Next, the post-processing pipeline converted the classification results to *BioC* format.

2.7 Heuristic Rules to Generate Concept Pairs

One of the critical challenges of relation extraction is that it has to consider the permutations among all clinical concepts

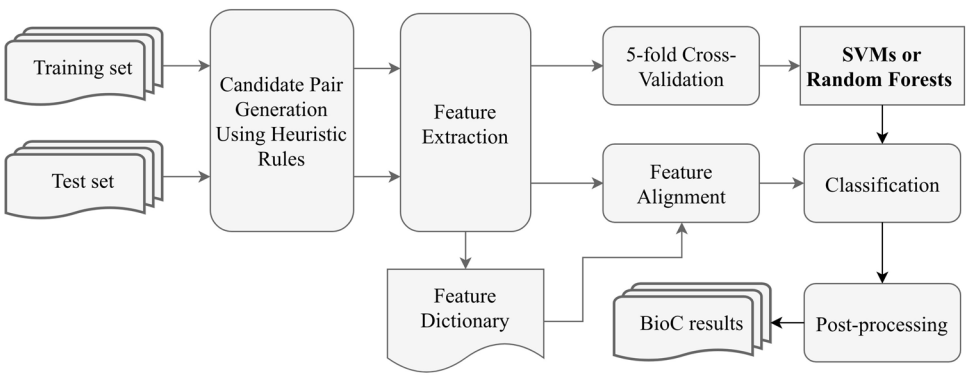

Fig. 2 Workflow of the relation extraction module. *SVMs* support vector machines

in the document level. However, this often brings too many negative samples, causing imbalanced positive/negative sample sizes. We developed the following heuristic rules to control the generation of candidate pairs:

Rule 1 Two clinical concepts occurring in the same sentence or two consecutive sentences will be considered as a candidate pair; continue to Rule 2, otherwise stop.

Rule 2 For each of the pairs generated in Rule 1, if the entity types of the two concepts fall into any possible combinations shown in Table 2, it will be a candidate pair for classification; otherwise stop.

We divided the candidate pairs generated by the heuristic rules into a set of single-sentence pairs and a set of cross-sentence pairs, and developed a single-sentence classifier and a cross-sentence classifier, respectively.

2.8 Handle Single-Sentence Relations and Cross-Sentence Relations

Since relations were annotated at the document level, they may occur within a single sentence or cross multiple sentences. We compared two relation extraction strategies, including (1) a one-classifier model for all relations; and (2) a separate-classifier model, i.e. one classifier for single-sentence relations and another classifier for cross-sentence relations.

2.9 Machine Learning-Based Relation Extraction Methods

We explored two machine learning methods to classify the relation types, including SVMs and RFs. These two machine learning models were widely used in various classification tasks and demonstrated good performance. We used the SVMs algorithm implemented in the LIBSVM-3.22 package [42] and the RFs algorithm implemented in the scikit-learn library (http://scikit-learn.org) The following features were extracted: (1) the local context information, including words inside each entity; (2) the distance between two clinical concepts in number of characters and number of words; (3) unigram, bigram, and trigram before and after each entity; (4) semantic information such as the entity types of the two entities in a relation and other entities that occurred in this sentence and their entity types. We optimized the features and parameters using grid searching based on fivefold cross-validation. For SVMs, we tuned the regularizer $\boldsymbol{c}$ and the tolerance of termination criterion $\boldsymbol{e}$. All other parameters were set as the default. For RFs, we tuned the number of trees (*n_estimators*) and the maximum features to include (*max_features*). The Gini impurity method was utilized as the tree splitting function.

2.9.1 Integrated Pipeline

We integrated the NER and relation extraction modules into a unified pipeline that can extract clinical concepts and their relations together from clinical text. In the integrated pipeline, the relation extraction module performs relation extraction based on clinical concepts detected by the NER module.

2.10 Experiments and Evaluation

Typically, the training of deep learning models requires a validation set to optimize parameters. Therefore, we divided the original training set of 876 notes into a short training set of 776 notes and a validation set of the remaining 100 notes. We trained an LSTM-CRFs model using the short-training set and optimized the parameters according to the performance on the validation set, denoted as RNN-1. We then combined the short training set and the validation set and retrained another LSTM-CRFs model according to the parameters optimized in RNN-1, denoted as RNN-2. In the training of LSTM-CRFs, we only used pre-trained word embeddings provided by the MADE1.0 organizers, without any feature engineering. The character embedding layer is randomly initialized and updated along the training progress. According to the performance on the validation set, the parameters of the LSTM-CRFs model were optimized as follows: the character embedding dimension was 25, the bidirectional word-level LSTM had an output dimension of 200, and the bi-directional character-level LSTM had an output size of 25; the learning rate fixed at 0.005; the input layer for the word-level LSTM applied a dropout at a probability of 0.5; and the stochastic gradient descending applied a gradient clapping at $[-5.0, 5.0]$. Using the optimized parameters, we trained another LSTM-CRFs model, RNN-2, using the entire training set (876 notes). For relation extraction, we optimized the SVMs using cross-validation. We excluded the bigram and trigram features for the RFs model as the cross-validation results showed a drop of 0.08 on the F1 score.

We used F1 score, precision, and recall to evaluate the performance of clinical NER and relation extraction. As both of these two tasks have multiple classes, we calculated the microaverage scores over all classes for evaluation. For clinical NER, we used the strict scores—both the offsets and the semantic type of a concept have to be exactly the same as those in the gold standard. All evaluation scores were calculated using the official evaluation scripts provided by the event organizer.

Table 3 The NER performances on both the validation and test sets

Model	Dataset	Performance		
		Precision	Recall	F1 score
CRFs	Validation	0.8555	0.8207	0.8377
RNN-1		**0.8893**	**0.8900**	**0.8897**
CRFs	Test	0.6618	0.8015	0.7250
RNN-1		0.8034	0.8236	0.8134
RNN-2		**0.8149**	**0.8318**	**0.8233**

CRFs conditional random fields, *RNN* recurrent neural network

The best scores on validation and test were highlighted in bold

Table 4 Performances of RNN-2 on the test set for each entity type

RNN-2	Performance		
Entity category	Precision	Recall	F1 score
Drug	0.8597	0.9003	0.8795
Indication	**0.5142**	**0.7605**	**0.6135**
Frequency	0.8467	0.8638	0.8552
Severity	0.7509	0.7832	0.7667
Dose	0.8815	0.8393	0.8592
Duration	0.6466	0.7890	0.7107
Route	0.8869	0.9274	0.9067
ADE	**0.5104**	**0.7432**	**0.6052**
SSLIF	0.8468	0.8319	0.8232

RNN recurrent neural network, *ADE* adverse drug event, *SSLIF* other signs, symptoms and diseases that are not an *ADE* or an *Indication*

The scores for indication and ADE were highlighted in bold, which are relatively lower than other entities

3 Results

3.1 Clinical NER

Table 3 shows the best performance of LSTM-CRFs and CRFs on the validation set of 100 notes and the test set of 213 notes. Using only the word embeddings provided by the organizer, the RNN-1 achieved the best F1 score of 0.8897, outperforming the CRFs model of 0.8377 on the validation set. We were unable to evaluate the RNN-2 model as the validation set was merged in order to train the RNN-2. On the test set, RNN-2 achieved the best F1 score of 0.8233, outperforming the RNN-1 of 0.8134 and the CRFs of 0.7250. Both RNN-1 and RNN-2 outperformed the baseline CRFs model by approximately 0.1 in terms of strict F1 score. The LSTM-CRFs model improved on both the precision and recall compared with CRFs.

Table 4 shows detailed evaluation scores of the best-performing NER model, RNN-2, for each entity type on the test data. The RNN-2 achieved good F1 scores for most of the nine entity types; however, the F1 scores (0.6135 and 0.6052, respectively) for *Indication* and *ADE* are lower than other entity types. Although the recalls of both *Indication* and *ADE* are decent (approximately 0.75), the precisions are notably low (approximately 0.5).

Table 5 Comparison between the one-classifier and separate-classifier for relation extraction on test data

Model	Performance		
	Precision	Recall	F1 score
One-classifier (SVMs)	0.8367	0.8242	0.8304
Separate-classifier (SVMs)	0.8491	0.8441	**0.8466**

SVMs support vector machines

The best F1 score was highlighted in bold

Table 6 Relation extraction performances on the training and the test sets using separate classifiers

Model	Expr.	Performance		
		Precision	Recall	F1 score
SVMs	Training	0.9199	0.9296	0.9247
RFs		0.9432	0.9289	**0.9360**
SVMs	Test	0.8491	0.8441	**0.8466**
RFs		0.8174	0.8505	0.8337

SVMs support vector machines, *RFs* random fields

Best scores on training and test were highlighted in bold

3.2 Relation Extraction

Using the heuristic rules, we derived a total of 64,783 single-sentence relation pairs, of which 18,948 (29.2%) were positive samples and 45,835 (70.8%) were negative samples. For cross-sentence relations, we only considered relations across two sentences as considering more than two sentences generated too many negative samples. Using the same heuristic rules, we derived a total of 31,406 cross-sentence relations, of which 2814 (9%) were positive samples and 28,592 (91%) were negative samples. We compared the one-classifier strategy, where a unified SVMs model was trained to handle all relations, with a separate-classifier strategy, where two SVMs models were trained, one for single-sentence relations and another for cross-sentence relations. The two models were optimized using fivefold cross-validation on the training set. Table 5 compares the performance of the two strategies on the test set. The separate-classifier outperformed the one-classifier. We then compared two machine learning models, including SVMs and RFs, using the separate-classifier strategy. Table 6 summarizes the best microaverage scores for the SVMs and RFs using both fivefold cross-validation on the training set and the final scores when applied

Table 7 Relation extraction performances for SVMs by relation type in the test set

SVM	Performance		
Relation type	Precision	Recall	F1 score
Severity_type	0.8766	0.9333	0.9041
Manner/route	0.9231	0.8660	0.8936
Reason	0.7546	0.8051	**0.7790**
do	0.9342	0.8803	0.9064
du	0.8979	0.6139	**0.7293**
fr	0.9096	0.9009	0.9052
Adverse	0.6774	0.7387	**0.7067**

SVM support vector machine, *do* dose relation between a drug and its dose, *du* duration, *fr* frequency

The F1 scores for reason, du, and adverse were highlighted in bold, which are relatively lower than other relations

Table 8 Performances of integrated task on the test set

Method	Performance		
	Precision	Recall	F1 score
RNN-2 + SVM	0.5758	0.6542	**0.6125**
RNN-2 + RF	0.5597	0.6543	0.6033

RNN recurrent neural network, *SVM* support vector machine, *RF* random field

The best F1 score was highlighted in bold

to the test. The RFs model achieved the best microaverage F1 score of 0.9360 using cross-validation on the training set, outperforming the SVMs model of 0.9247; however, the SVMs model achieved the best microaverage F1 score of 0.8466 on the test set.

Table 7 shows the precision, recall, and F1 score of the SVMs model for each relation type in the test set. The SVMs model achieved good F1 scores (approximately 0.9) for the severity_type, manner/router, do, and fr categories, yet the F1 scores for the reason, du, and adverse categories are relatively lower (between 0.7 and 0.8).

3.3 The Integrated System

As the performance of SVMs and RFs are comparable for relation extraction, we developed the integrated pipeline by integrating the best NER model, RNN-2, with both of the two relation extraction methods—RNN-2 + SVMs and RNN-2 + RFs. Table 8 shows the performance of the two pipelines. The RNN-2 + SVMs pipeline achieved a better F1 score of 0.6125, outperforming the RNN-2 + RFs pipeline of 0.6033. The experimental result is consistent with relation extraction, where the SVMs model is better than RFs.

4 Discussion

Early detection and prevention of ADEs is important for a safer and higher-quality healthcare. A prerequisite of using narrative clinical text for early prevention of ADEs is to identify mentions of medications, ADEs, and their relations. In this study, we presented MADEx, an NLP system to detect medications, ADEs, and their relations from clinical notes. We applied a state-of-the-art method (LSTM-CRFs) for clinical NER and compared it with a traditional machine learning method (CRFs). The best LSTM-CRFs model (RNN-2) achieved a microaverage F1 score of 0.8233, outperforming the baseline CRFs model. According to the official evaluation results from organizers, our system achieved a top-three best performance among ten participating teams, and 23 submitted runs in the NER task of the MADE1.0 open challenge. Our results demonstrated the superior performance of the LSTM-CRFs model for clinical NER. We also developed a hybrid relation extraction module using SVMs and compared it with another widely used machine learning model, RFs. We then integrated the best-performing NER module, RNN-2 with an SVMs-based relation extraction module, into an integrated pipeline. The post-challenge evaluation showed that the SVMs-based relation extraction module achieved a microaverage F1 score of 0.8466, outperforming an RFs-based relation extraction method; the integrated system, RNN-2 + SVMs, achieved a microaverage F1 score of 0.6125. The relation extraction module and the integrated pipeline of MADEx are comparable to the best-performing systems in this challenge (0.8684 and 0.6170, respectively).

4.1 The NER Task

For the NER task, LSTM-CRFs outperformed the baseline CRFs (F1 score of 0.8233 vs. 0.7250) using only word embeddings provided by the organizer, demonstrating the efficiency of LSTM-CRFs for clinical NER. Compared with the baseline CRFs, the LSTM-CRFs improved both precision (0.8149 vs. 0.6618) and recall (0.8318 vs. 0.8015). The baseline CRFs utilized human-generated features that were reported to be useful for NER in previous studies; however, the LSTM-CRFs only utilized word embeddings, which are numeric vectors trained from large unlabeled medical text without human intervention. The organizers trained this word embedding from three resources, including the English Wikipedia, a set of 99,700 electronic health record notes, and PubMed open access articles [27]. Deep learning models have a promise to automatically learn high-level feature representations in an unsupervised manner. Many studies [41, 43] have shown that word embeddings trained from a large corpus can capture multi-aspect semantic knowledge to

improve the performance of clinical NER. Our findings are consistent with previous studies and further demonstrated the advantage of LSTM-CRFs to utilize large, unlabeled corpus for clinical NER.

We developed a new modified training strategy for LSTM-CRFs in this study. The typical training of a deep learning model requires a validation set to optimize the parameters. During training, researchers use the performance on the validation set to evaluate the models generated at different training iterations, choose the stop point, and select the best models for testing. However, there is often no validation set provided in some real-life applications, such as this challenge. A typical solution is to split a proportion of notes from the original training as validation. Thus, the original training becomes a short-training set. This study proposed and examined a new training strategy; after optimizing the parameters using the short training, we merged the validation set into the short training set and retrained a new model using the optimized parameters. The traditional training strategy was able to find a local maximum but has less training samples, whereas our new training strategy was able to explore more training samples, but the final model might not be at a local maximum. The experimental results showed that the RNN-2 model (F1 score of 0.8233), trained using the new strategy, outperformed the RNN-1 model (F1 score of 0.8134), trained using the traditional short training, indicating that having more samples in training may be more important than finding a local maximum. Further investigation should examine the effect of corpus size and local maximum selection.

4.2 The Relation Extraction Task

Relation extraction is a critical NLP task to understand the relations between clinical concepts. We developed an SVMs-based relation extraction module and compared it with RFs. Two SVMs-based models were developed for the single-sentence pairs and the cross-sentence pairs. As shown in Table 6, RFs achieve a better F1 score than SVMs using cross-validation on the training. The performance of both SVMs and RFs dropped when they were applied to the test set. SVMs outperformed the RFs on the final test set by >0.01 in terms of F1 score, indicating that SVMs may be more generalizable than the RFs for relation extraction. Further investigation should examine the differences.

We were not able to finish the relation extraction module and the integrated pipeline during the challenge. The post-challenge evaluation using the official evaluation scripts showed that the relation extraction module of MADEx outperformed the second-best system (approximately 0.02 lower than the best system [44] on F1 score) on relation extraction task, and our integrated system, RNN-2 + SVMs, outperformed the second-best system on integrated task (0.0045 lower than the best system [45]). Our MADEx system is comparable to the best-performing systems for relation extraction and integrated task in this challenge. The top-performing system for the integrated task in this challenge achieved a microaverage F1 score of approximately 0.61, indicating that the integrated task of extracting entity and relation together from clinical text remains a challenging problem. Relation extraction is challenging for several reasons. First, relations are annotated at document level and there are relations annotated across multiple sentences. Thus, the relation extraction systems have to consider the combinations between all clinical concepts within a document. Second, compared with the word-level applications such as clinical NER, the features for relation extraction are not that straightforward. Similar to our previous studies on relation extraction [34], we developed heuristic rules to generate candidate pairs to control the ratio between negative and positive samples. For cross-sentence relations, MADEx only considered relations within two consecutive sentences. Although it excluded the relations across more than two sentences, this strategy provided a reasonable positive/negative sample ratio. We also tried to include the relations across more than two sentences, however it brought more noise to the training set and caused a serious imbalance issue.

4.3 Error Analysis and Future Work

As shown in Table 4, the performance scores of NER are notably lower for *ADE* and *Indication* compared with other entity types. We analyzed errors for the two entity types. Some false negatives were caused by boundary mismatching or misclassification of semantic types. For example, a common type of error for *ADE* and *Indication* is to misclassify them as SSLIF (other signs, symptoms, and diseases that are not an *ADE* or an *Indication*). This may be caused by the limited number of training samples (*ADE* and *Indication* only accounted for approximately 2% and 5% of the total number of entities, respectively). Some entities were annotated with two different semantic categories and our NER module could not handle them correctly. For example, the entity 'attention and concentration span has decreased' is annotated both as *ADE* and *SSLIF*. There were also complex entities such as 'nodular-sclerosing stage IIa Hodgkin disease', which was annotated as *Indication*, and part of it, 'stage IIa', was annotated as *Severity*. Our NER module was able to extract entities 'nodular-sclerosing' as *SSLIF*, 'stage IIa' as *Severity*, and 'Hodgkin disease' as *Indication*, but failed to detect the whole sequence as an *Indication*.

For relation extraction task, the *adverse*, *reason*, and *du* categories have notably lower F1 scores compared with other relation types, as shown in Table 6. Part of the reason for this is that clinical concepts from the 'adverse' category are very similar to clinical concepts from 'reason' category.

Since the entities of *ADE*, *Indication*, and *SSLIF* tend to have similar contexts, it is hard to discriminate the relation types using only the contextual information. For the 'du' category, the limited number of training samples was the main problem ('du' accounted for only 4% of the total relations).

We will continue to improve the performance of MADEx in our future work. To improve the performance of detecting *ADE* and *Indication* concepts, we plan to develop new methods to integrate medical knowledge with corpus-based word embeddings to help distinguish between the two categories [46]. We also plan to design post-processing rules to improve the NER module for detecting complex entities. The new methods to integrate medical knowledge with corpus-based word embedding will also help distinguish among different relation types. The CNNs have demonstrated good performance for relation extraction, which is our next focus. We will also explore the joint learning models that perform NER and relation extraction in a unified model.

5 Conclusions

In this study, we presented MADEx, a machine learning-based NLP system to detect medications, ADEs, and their relations from clinical text. MADEx consists of a clinical NER module implemented using LSTM-CRFs and a relation extraction module implemented using SVMs. MADEx achieved top-three best performance on the NER task of the MADE1.0 challenge, demonstrating the efficiency of LSTM-CRFs for clinical NER. The post-challenge evaluation showed that the relation extraction module and integrated pipeline of MADEx are comparable to the best-performing systems developed in this challenge.

Acknowledgements The authors would like to thank the organizers who provided the annotated corpus and word embeddings for this challenge, and gratefully acknowledge the support of the NVIDIA Corporation with the donation of the GPUs used for this research. The authors would also like to thank the anonymous reviewers for their helpful feedback.

Compliance with Ethical Standards

Funding This study was supported in part by the University of Florida Clinical and Translational Science Institute, which is funded by the National Institutes of Health (NIH) National Center for Advancing Translational Sciences under award number UL1TR001427, and the OneFlorida Clinical Research Consortium, which is funded by the Patient-Centered Outcomes Research Institute (PCORI) under award number CDRN-1501-26692. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

Conflict of Interest Xi Yang, Jiang Bian, Yan Gong, William R. Hogan, and Yonghui Wu have no conflicts of interest to declare that are directly relevant to the contents of this study.

Ethical Considerations This study utilized de-identified clinical notes provided by the University of Massachusetts Medical School through the MADE1.0 challenge, and was approved by the University of Florida Institutional Review Board.

References

1. Institute of Medicine (US) Committee on quality of health care in America. To err is human: building a safer health system. Washington, DC: National Academies Press; 2000. http://www.ncbi.nlm.nih.gov/books/NBK225182/. Accessed 23 June 2018.
2. Weiss AJ, Freeman WJ, Heslin KC, Barrett ML. Adverse drug events in US Hospitals, 2010 versus 2014. Statistical brief #234. AHRQ; 2018. https://www.hcup-us.ahrq.gov/reports/statbriefs/sb234-Adverse-Drug-Events.jsp. Accessed Dec 2018.
3. Stausberg J. International prevalence of adverse drug events in hospitals: an analysis of routine data from England, Germany, and the USA. BMC Health Serv Res. 2014;14:125.
4. Poudel DR, Acharya P, Ghimire S, Dhital R, Bharati R. Burden of hospitalizations related to adverse drug events in the USA: a retrospective analysis from large inpatient database. Pharmacoepidemiol Drug Saf. 2017;26:635–41.
5. Aljadhey H, Mahmoud MA, Mayet A, Alshaikh M, Ahmed Y, Murray MD, et al. Incidence of adverse drug events in an academic hospital: a prospective cohort study. Int J Qual Health Care. 2013;25:648–55.
6. Aljadhey H, Mahmoud MA, Ahmed Y, et al. Incidence of adverse drug events in public and private hospitals in Riyadh, Saudi Arabia: the (ADESA) prospective cohort study. BMJ Open. 2016;6:e010831.
7. Wang Y, Wang L, Rastegar-Mojarad M, et al. Clinical information extraction applications: a literature review. J Biomed Inform. 2018;77:34–49.
8. Meystre SM, Savova GK, Kipper-Schuler KC, Hurdle JF. Extracting information from textual documents in the electronic health record: a review of recent research. Yearb Med Inform. 2008;17:128–44.
9. Nadkarni PM, Ohno-Machado L, Chapman WW. Natural language processing: an introduction. J Am Med Inform Assoc. 2011;18:544–51.
10. Kumar S. A survey of deep learning methods for relation extraction; 2017. arXiv:170503645.
11. Uzuner Ö, South BR, Shen S, DuVall SL. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. J Am Med Inform Assoc. 2011;18:552–6.
12. Sun W, Rumshisky A, Uzuner O. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. J Am Med Inform Assoc. 2013;20:806–13.
13. Pradhan S, Elhadad N, Chapman W, Manandhar S, Savova G. SemEval-2014 Task 7: analysis of clinical text. In: Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014);2014. p. 54–62.
14. Pradhan S, Elhadad N, South BR, Martinez D, Christensen L, Vogel A, et al. Evaluating the state of the art in disorder recognition and normalization of the clinical narrative. J Am Med Inform Assoc. 2015;22:143–54.
15. Lafferty JD, McCallum A, Pereira FCN. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th international conference on machine learning. San Franciso, CA: Morgan Kaufmann Publishers Inc.; 2001. p. 282–89.
16. Hearst MA, Dumais ST, Osuna E, Platt J, Scholkopf B. Support vector machines. IEEE Intell Syst Appl. 1998;13:18–28.

17. Tsochantaridis I, Joachims T, Hofmann T, Altun Y. Large margin methods for structured and interdependent output variables. J Mach Learn Res. 2005;6:1453–84.
18. Aronson AR, Lang F-M. An overview of MetaMap: historical perspective and recent advances. J Am Med Inform Assoc. 2010;17:229–36.
19. Friedman C. Towards a comprehensive medical language processing system: methods and issues. Proc AMIA Annu Fall Symp. 1997;595–599.
20. Denny JC, Irani PR, Wehbe FH, Smithers JD, Spickard A. The KnowledgeMap project: development of a concept-based medical school curriculum database. AMIA Annu Symp Proc.; 2003. pp. 195–199.
21. de Bruijn B, Cherry C, Kiritchenko S, Martin J, Zhu X. Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010. J Am Med Inform Assoc. 2011;18:557–62.
22. Zhang Y, Wang J, Tang B, Wu Y, Jiang M, Chen Y, et al. UTH_CCB: a report for semeval 2014–task 7 analysis of clinical text. Sem Eval. 2014;2014:802.
23. Tang B, Wu Y, Jiang M, Denny JC, Xu H. Recognizing and encoding disorder concepts in clinical text using machine learning and vector space model. CLEF 2013 proceedings. 2013. http://ceur-ws.org/Vol-1179/CLEF2013wn-CLEFeHealth-TangEt2013.pdf.
24. Le H-Q, Nguyen TM, Vu ST, Dang TH. D3NER: biomedical named entity recognition using CRF-biLSTM improved with fine-tuned embeddings of various linguistic information. Bioinformatics. 2018;24(20):3539–46.
25. Habibi M, Weber L, Neves M, Wiegandt DL, Leser U. Deep learning with word embeddings improves biomedical named entity recognition. Bioinformatics. 2017;33:i37–48.
26. Liu Z, Yang M, Wang X, Chen Q, Tang B, Wang Z, et al. Entity recognition from clinical texts via recurrent neural network. BMC Med Inform Decis Mak. 2017;17(Suppl 2):67.
27. Jagannatha AN, Yu H. Bidirectional RNN for medical event detection in electronic health records. Proc Conf. 2016;2016:473–82.
28. Wu Y, Jiang M, Lei J, Xu H. Named entity recognition in chinese clinical text using deep neural network. Stud Health Technol Inform. 2015;216:624–8.
29. Wu Y, Jiang M, Xu J, Zhi D, Xu H. Clinical named entity recognition using deep learning models. AMIA Annu Symp Proc 2018; 2017:1812–19 (**eCollection 2017**).
30. Zhao S, Grishman R. Extracting relations with integrated information using Kernel methods. In: Proceedings of the 43rd annual meeting of the association for computational linguistics. Stroudsburg, PA; 2005. pp. 419–426.
31. Tikk D, Thomas P, Palaga P, Hakenberg J, Leser U. A comprehensive benchmark of kernel methods to extract protein-protein interactions from literature. PLoS Comput Biol. 2010;6:e1000837.
32. Zelenko D, Aone C, Richardella A. Kernel methods for relation extraction. J Mach Learn Res. 2003;3:1083–106.
33. Brin S. Extracting patterns and relations from the world wide web. In: Atzeni P, Mendelzon A, Mecca G, editors. The world wide web and databases. London: Springer; 1999. p. 172–83.
34. Tang B, Wu Y, Jiang M, Chen Y, Denny JC, Xu H. A hybrid system for temporal information extraction from clinical text. J Am Med Inform Assoc. 2013;20:828–35.
35. Xu J, Wu Y, Zhang Y, Wang J, Lee H-J, Xu H. CD-REST: a system for extracting chemical-induced disease relation in literature. Database. 2016;2016:baw036.
36. Wei C-H, Peng Y, Leaman R, Davis AP, Mattingly CJ, Li J, et al. Assessing the state of the art in biomedical relation extraction: overview of the BioCreative V chemical-disease relation (CDR) task. Database. 2016;2016:baw032.
37. Comeau DC, Islamaj Doğan R, Ciccarese P, et al. BioC: a minimalist approach to interoperability for biomedical text processing. Database. 2013;2013:bat064.
38. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9:1735–80.
39. Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C. Neural architectures for named entity recognition; 2016. arXiv :160301360.
40. Soysal E, Wang J, Jiang M, Wu Y, Pakhomov S, Liu H. CLAMP—a toolkit for efficiently building customized clinical natural language processing pipelines. J Am Med Inform Assoc. 2017. https://doi.org/10.1093/jamia/ocx132.
41. Wu Y, Xu J, Jiang M, Zhang Y, Xu H. A study of neural word embeddings for named entity recognition in clinical text. AMIA Annu Symp Proc. 2015;2015:1326–33.
42. LIBSVM. A library for support vector machines. https://www.csie.ntu.edu.tw/~cjlin/libsvm/. Accessed 23 Jun 2018.
43. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. J Mach Learn Res. 2011;12:2493–537.
44. Chapman AB, Peterson KS, Alba PR, DuVall SL, Patterson OV. Hybrid system for adverse drug event detection. Proc Mach Learn Res. 2018;90:16–24.
45. Dandala B, Joopudi V, Devarakonda M. IBM Research System at MADE 2018: detecting adverse drug events from electronic health records. Proc Mach Learn Res. 2018;90:39–47.
46. Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell. 2013;35:1798–828.