Bidirectional RNN for Medical Event Detection in Electronic Health Records

Abhyuday N Jagannatha[1] and Hong Yu[1,2]

[1]University of Massachusetts, MA, USA

[2]Bedford VAMC and CHOIR, MA, USA

Abhyuday N Jagannatha: abhyuday@cs.umass.edu; Hong Yu: hong.yu@umassmed.edu

Abstract

Sequence labeling for extraction of medical events and their attributes from unstructured text in Electronic Health Record (EHR) notes is a key step towards semantic understanding of EHRs. It has important applications in health informatics including pharmacovigilance and drug surveillance. The state of the art supervised machine learning models in this domain are based on Conditional Random Fields (CRFs) with features calculated from fixed context windows. In this application, we explored recurrent neural network frameworks[1] and show that they significantly out-performed the CRF models.

1 Introduction

EHRs report patient's health, medical history and treatments compiled by medical staff at hospitals. It is well known that EHR notes contain information about medical events including medication, diagnosis (or Indication), and adverse drug events (ADEs) etc. A medical event in this context can be described as a change in patient's medical status. Identifying these events in a structured manner has many important clinical applications such as discovery of abnormally high rate of adverse reaction events to a particular drug, surveillance of drug efficacy, etc. In this paper we treat EHR clinical event detection as a task of sequence labeling.

Sequence labeling in the context of machine learning refers to the task of learning to predict a label for each data-point in a sequence of data-points. This learning framework has wide applications in many disciplines such as genomics, intrusion detection, natural language processing, speech recognition etc. However, sequence labeling in EHRs is a challenging task. Unlike text in the open domain, EHR notes are frequently noisy, containing incomplete sentences, phrases and irregular use of language. In addition, EHR notes incorporate abundant abbreviations, rich medical jargons, and their variations, which make recognizing semantically similar patterns in EHR notes difficult. Additionally, different events exhibit different patterns and possess different prevalences. For example, while a medication comprises of at most a few words of a noun, an ADE (e.g., "has not felt back to his normal self") may vary to comprise of a significant part of a sentence. While medication information is frequently described in EHRs, ADEs are typically rare events.

Rule-based and learning-based approaches have been developed to identify and extract information from EHR notes (Haerian et al., 2012), (Xu et al., 2010), (Friedman et al., 1994), (Aronson, 2001), (Polepalli Ramesh et al., 2014). Learning-based approaches use sequence labeling algorithms like Conditional Random Fields (Lafferty et al., 2001), Hidden Markov Models (Collier et al., 2000), and Max-entropy Markov Models (McCallum et al., 2000). One major drawback of these graphical models is that the label prediction at any time point only depends on its data instance and the immediate neighboring labels.

While this approach performs well in learning the distribution of the output labels, it has some limitations. One major limitation is that it is not designed to learn from dependencies which lie in the surrounding but not quite immediate neighborhood. Therefore, the feature vectors have to be explicitly modeled to include the surrounding contextual information. Traditionally, bag of words representation of surrounding context has shown reasonably good performance. However, the information contained in the bag of words vector is very sensitive to context window size. If the context window is too short, it will not include all the information. On the other hand if the context window is too large, it will compress the vital information with other irrelevant words. Usually a way to tackle this problem is to try different context window sizes and use the one that gives the highest validation performance. However, this method cannot be easily applied to our task, because different medical events like medication, diagnosis or adverse drug reaction require different context window sizes. For example, while a medication can be determined by a context of two or three words containing the drug name, an adverse drug reaction would require the context of the entire sentence. As an example, this is a sentence from one of the EHRs, "The follow-up needle biopsy results were consistent with *bronchiolitis obliterans*, which was likely due to the Bleomycin component of his *ABVD chemo*". In this sentence, the true labels are *Adverse Drug Event(ADE)* for "*bronchiolitis obliterans*" and *Drugname* for "*ABVD chemo*". However the ADE, "*bronchiolitis obliterans*" could be misslabeled as just another disease or symptom, if the entire sentence is not taken into context.

Recent advancements in Recurrent Neural Networks (RNNs) have opened up new avenues of research in sequence labeling. Traditionally, recurrent neural networks have been hard to train through Back-Propagation, because learning long term dependencies using simple recurrent neurons lead to problems like exploding or vanishing gradients (Bengio et al., 1994), (Hochreiter et al., 2001). Recent approaches have modified the simple neuron structure in order to learn dependencies over longer intervals more efficiently. In this study, we evaluate the performance of two such neural networks, namely, Long Short Term Memory (LSTM) and Gated Recurrent Units (GRU).

Timely identification of new drug toxicities is an unresolved clinical and public health problem, costing people's lives and billions of US dollars. In this study, we empirically evaluated LSTM and GRU on EHR notes, focusing on the clinically important task of detecting medication, diagnosis, and adverse drug event. To our knowledge, we are the first group reporting the uses of RNN frameworks for information extraction in EHR notes.

2 Related Work

Medication and ADE detection is an important NLP task in biomedicine. Related existing NLP approaches can be grouped into knowledge or rule-based, supervised machine learning, and hybrid approaches. For example, Hazlehurst et al. (2005) developed MediClass, a knowledge-based system that deploys a set of domain-specific logical rules for medical concept extraction. Wang et al. (2015), Humphreys et.al. (1993) and others map EHR notes to medical concepts to an external knowledge resource using hybrid rule-based and syntactic parsing approaches. Gurulingappa et al. (2010) detect two medical entities (*disease* and *adverse events*) in a corpus of annotated Medline abstracts. In contrast, our work uses a corpus of actual medical notes and detects additional events and attributes.

Rochefort et al. (2015) developed document classifiers to classify whether a clinical note contains deep venous thromboembolisms and pulmonary embolism. Haerian et al. (2012) applied distance supervision to identify terms (e.g., including "suicidal", "self harm", and "diphenhydramine overdose") associated with suicide events. Zuofeng Li et al. (2010) extracted medication information using CRFs.

Many named entity recognition systems in the biomedical domain have been driven by the Shared tasks of BioNLP (Kim et al., 2009), BioCreAtivE (Hirschman et al., 2005) i2b2 shared NLP tasks (Li et al., 2009) and ShARe/CLEF evaluation tasks (Pradhan et al., 2014). The best performing clinical NLP systems for named entity recognition includes Tang et al (2013) which applied CRF and structured SVM.

Neural Network models like Convolutional Neural Networks and Recurrent Neural Networks (LSTM, GRU) have recently been been successfully used to tackle various sequence labeling problems in NLP. Collobert (2011) used Convolutional Neural Network for sequence labeling problems like POS tagging, NER etc. Later, Huang et al. (2015) achieved comparable or better scores using bi-directional LSTM based models.

3 Dataset

The annotated corpus contains 780 English EHR notes or 613,593 word tokens (an average of 786 words per note) from cancer patients who have been diagnosed with hematological malignancy. Each note was annotated by at least two annotators with inter-annotator agreement of 0.93 kappa. The annotated events and attributes and their instances in the annotated corpus are shown in Table 1.

Table 1

Annotation statistics for the corpus.

Labels	Annotations	Avg. Words/Annotations
ADE	905	1.51
Indication	1988	2.34
Other SSD	26013	2.14
Severity	1928	1.38
Drugname	9917	1.20
Duration	562	2.17
Dosage	3284	2.14
Route	1810	1.14
Frequency	2801	2.35

The annotated events can be broadly divided into two groups, *Medication*, and *Disease*. The *Medication* group contains *Drugname*, *Dosage*, *Frequency*, *Duration* and *Route*. It corresponds to information about medication events and their attributes. The attributes (Route, Frequency, Dosage, and Duration) of a medication (Drug name) occur less frequently than the Drugname tag itself, because few EHRs report complete attributes of an event.

The *Disease* group contains events related to diseases (*ADE*, *Indication*, *Other SSD*) and their attributes (*Severity*). An injury or disease can be labeled as *ADE*, *Indication*, or *Other SSD* depending on the semantic context. It is marked as *ADE* if it is the side effect of a drug. It is marked as *Indication* if it is being diagnosed currently by the doctor and a medication has been prescribed for it. Any sign, symptom or disease that does not fall into the aforementioned two categories is labeled as *Other SSD*. *Other SSD* is the most common label in our corpus, because it is frequently used to label conditions in the past history of the patient.

For each note, we removed special characters that do not serve as punctuation and then split the note into sentences using regular expressions.

4 Methods

4.1 Long Short Term Memory

Long Short Term Memory Networks ([Hochreiter and Schmidhuber, 1997](#)) are a type of Recurrent Neural Networks (RNNs). RNNs are modifications of feed-forward neural networks with recurrent connections. In a typical NN, the neuron output at time t is given by:

$$y_i^t = \sigma\left(W_i x_t + b_i\right) \tag{1}$$

Where W_i is the weight matrix, b_i is the bias term and σ is the sigmoid activation function. In an RNN, the output of the neuron at time $t-1$ is fed back into the neuron. The new activation function now becomes:

$$y_i^t = \sigma\left(W_i x_t + U_i y_i^{t-1} + b_i\right) \tag{2}$$

Since these RNNs use the previous outputs as recurrent connections, their current output depends on the previous states. This property remembers previous information about the sequence, making them useful for sequence labeling tasks. RNNs can be trained through back-propagation through time. [Bengio et al. (1994)](#) showed that learning long term dependencies in recurrent neural networks through gradient decent is difficult. This is mainly because the back-propagating error can frequently "blow-up" or explode which makes convergence infeasible, or it can vanish which renders the network incapable of learning long term dependencies ([Hochreiter et al., 2001](#)).

In contrast, LSTM networks were proposed as solutions for the vanishing gradient problem and were designed to efficiently learn long term dependencies. LSTMs accomplish this by keeping an internal state that represents the memory cell of the LSTM neuron. This internal state can only be read and written through gates which control the information flowing through the cell state. The updates of various gates can be computed as:

$$i_t = \tanh(W_{xi}x_t + W_{hi}h_{t-1}) \tag{3}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1}) \tag{4}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1}) \tag{5}$$

Here i_t, f_t and o_t denote input, forget and output gate respectively. The forget and input gate determine the contributions of the previous output and the current input, in the new cell state c_t. The output gate controls how much of c_t is exposed as the output. The new cell state c_t and the output h_t can be calculated as follows:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc} x_t + W_{hc} h_{t-1}) \tag{6}$$

$$h_t = o_t \odot \tanh(c_t) \tag{7}$$

The cell state stores relevant information from the previous time-steps. It can only be modified in an additive fashion via the input and forget gates. Simplistically, this can be viewed as allowing the error to flow back through the cell state unchecked till it back propagates to the time-step that added the relevant information. This nature allows LSTM to learn long term dependencies.

We use LSTM cells in the Neural Network setup shown in figure 1. Here x_k, y_k are the input word, and the predicted label for the kth word in the sentence. The embedding layer contains the word vector mapping from words to dense n-dimensional vector representations. We initialize the embedding layer at the start of the training with word vectors calculated on the larger data corpus described in section 4.4. This ensures that words which are not seen frequently in the labeled data corpus still have a reasonable vector representation. This step is necessary because our unlabeled corpus is much larger than the labeled one.
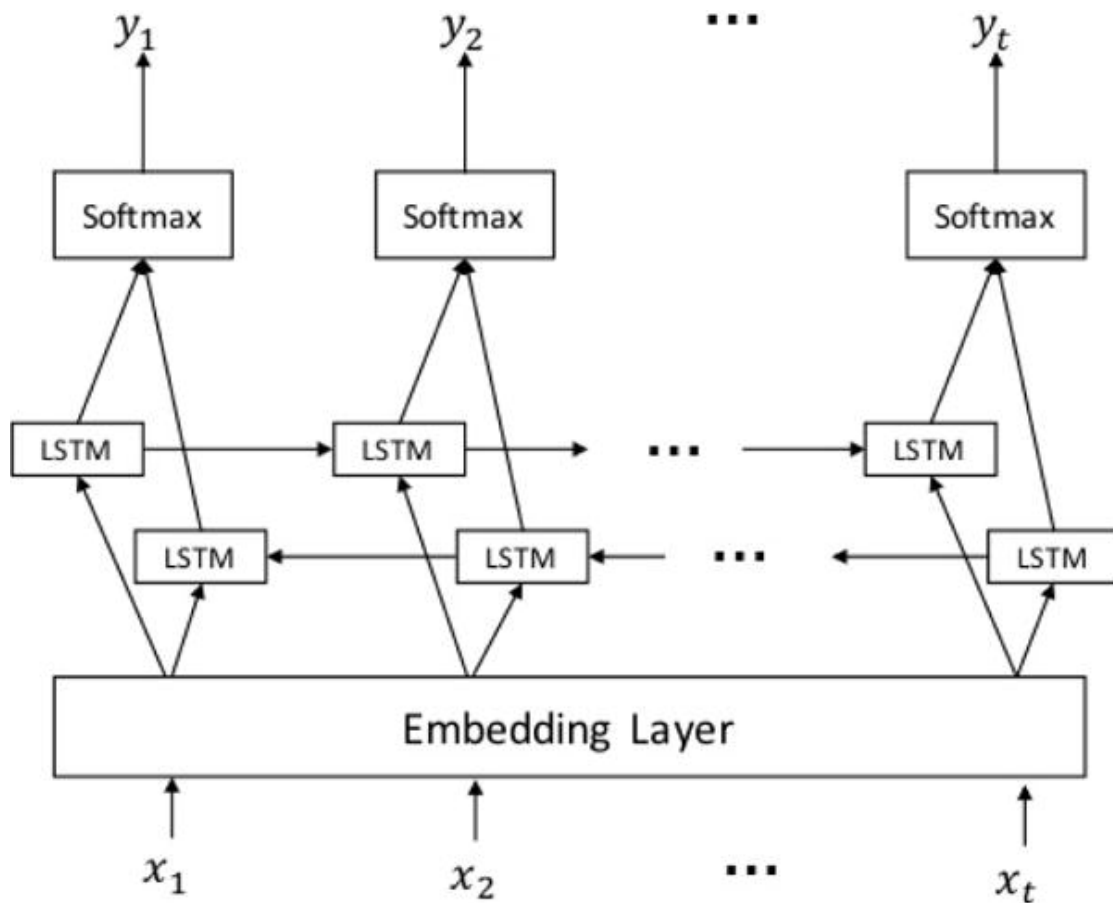
Figure 1

Sequence Labeling model for LSTM network

The words are mapped into their corresponding vector representations and fed into the LSTM layer. The LSTM layer consists of two LSTM chains, one propagating in the forward direction and other in the backward direction. We concatenate the output from the two chains to form a combined representation of the word and its context. This concatenated vector is then fed into a feed-forward neuron with Softmax activation function. The Softmax activation function normalizes the outputs to produce probability like outputs for each label type j as follows:

$$P\left(l_t = j | u_t\right) = \frac{exp\left(u_t W_j\right)}{\sum_{k=1}^{K} exp\left(u_t W_k\right)} \tag{8}$$

Here l_t and u_t are the label and the concatenated vector for each time step t. The most likely label at each word position is selected. The entire network is trained through back-propagation. The embedding vectors are also updated based on the back-propagated errors.

4.2 Gated Recurrent Units

Gated Recurrent Unit (GRU) is another type of recurrent neural network which was recently proposed for the purposes of Machine Translation by [Cho et. al. (2014)](#). Similar to LSTMs, Gated Recurrent Units also have an additive mechanism to update the cell state, with the current update. However, GRUs have a different mechanism to create the update. The candidate activation $\widetilde{h}_t$ is computed based on the previous cell state and the current input.

$$\widetilde{h}_t = \sigma \left(W_{xh} x_t + W_{hh} \left(r_t \odot h_{t-1} \right) \right) \tag{9}$$

Here r_t is the reset gate and it controls the use of previous cell state while calculating the input activation. The reset gate itself is also computed based on the previous cell activation h_{t-1} and the current candidate activation.

$$\widetilde{r}_t = \sigma \left(W_{xr} x_t + W_{hr} h_{t-1} \right) \tag{10}$$

The current cell state or activation is a linear combination of previous cell activation and the candidate activation.

$$h_t = \left(1 - z_t \right) \odot h_{t-1} + z_t \odot \widetilde{h}_t \tag{11}$$

Here, z_t is the update gate which decides how much contribution the candidate activation and the previous cell state should have in the cell activation. The update gate is computed using the following equation:

$$z_t = (W_{hz} h_{t-1} + W_{xz} x_t) \tag{12}$$

Gated recurrent units have some fundamental differences with LSTM. For example, there is no mechanism like the output gate which controls the exposure of the cell activation, instead the entire current cell activation is used as output. The mechanisms for using the previous output for the calculation of the current activation are also very different. Recent experiments ([Chung et al., 2014](#)), ([Jozefowicz et al., 2015](#)) comparing both these architectures have shown GRUs to have comparable or sometimes better performance than LSTM in several tasks with long term dependencies.

We use GRU with the same Neural Network structure as shown in [Figure 1](#) by replacing the LSTM nodes with GRU. The embedding layer used here is also initialized in a similar fashion as the LSTM network.

4.3 The Baseline System

CRFs have been widely used for sequence labeling tasks in NLP. CRFs model the complex dependence of the outputs in a sequence using Probabilistic Graphical Models. Probabilistic Graphical Models represent relationships between variables through a product of factors where each factor is only influenced by a smaller subset of the variables. A particular factorization of the variables provides a specific set of independence relations enforced on the data. Unlike Hidden Markov Models which model the joint $p(x, y)$, CRFs model the posterior probability $p(y|x)$ directly. The conditional can be written as a product of factors as follows:

$$p(y|x) = \frac{1}{Z(x)} \prod_{t=1}^{T} \psi_t(y_t, y_{t-1}, x_t) \tag{13}$$

Here Z is the partition function used for normalization, ψ_t are the local factor functions.

CRFs are fed the word inputs and their corresponding skip-gram word embedding (section 4.4). To compare CRFs with RNN, we add extra context feature for each word. This is done because our aim is to show that RNNs perform better than CRFs using context windows. This extra feature consists of two vectors that are bag of words representation of the sentence sections before and after the word respectively. We add this feature to explicitly provide a mechanism that is somewhat similar to the surrounding context that is generated in a Bi-directional RNN as shown in Figure 1. This CRF model is referred to as CRF-context in our paper. We also evaluate a CRF-nocontext model, which trains a CRF without the context features.

The tagging scheme used with both CRF models is BIO (Begin, Inside and Outside). We did not use the more detailed BILOU scheme (Begin, Inside, Last, Outside, Unit) due to data sparsity in some of the rarer labels.

4.4 Skip-Gram Word Embeddings

We use skip-gram word embeddings trained through a shallow neural network as shown by Mikolov et al., (2013) to initialize the embedding layer of the RNNs. This embedding is also used in the baseline CRF model as a feature. The embeddings are trained on a large unlabeled biomedical dataset, compiled from three sources, the English Wikipedia, an unlabeled EHR corpus, and PubMed Open Access articles. The English Wikipedia consists of text extracted from all the articles of English Wikipedia 2015. The unlabeled EHR corpus contains 99,700 electronic health record notes. PubMed Open Access articles are obtained by extracting the raw text from all openly available PubMed articles. This combined raw text corpus contains more than 3 billion word tokens. We convert all words to lowercase and use a context window of 10 words to train a 200 dimensional skip gram word embedding.

5 Experiments and Evaluation Metrics

For each word, the models were trained to predict either one of the nine medically relevant tags described in section 3, or the *Outside* label. The CRF tagger was run in two modes. The first mode (CRF– nocontext) used only the current word and its corresponding skip-gram representation. The second mode (CRF– context) used the extra context feature described in section 4.3. The extra features are basically the bag of words representation of the preceding and following sections of the sentence. The first mode was used to compare the performance of CRF and RNN models when using the same input data. It also serves as a method of contrasting with CRF's performance when context features are explicitly added. CRF Tagger uses L-BFGS optimizer with L2- regularization.

The RNN frameworks are trained on sentence level and document level. The sentence level neural networks are fed only one sentence at a time. This means that the LSTM and GRU states are only preserved and propagated within a sentence. The networks cell states are re-initialized before each sentence. The document level neural networks are fed one document at a time, so they can learn context cues that reside outside of the sentence boundary. We use 100 dimensional hidden layer for each directional RNN chain. Since we use bi-directional LSTMs and GRUs, this essentially amounts to a 200 dimensional recurrent hidden layer. The hidden layer activation functions for both RNN models are *tanh*. Output of this hidden layer is fed into a Soft-max output layer which emits probabilities for each of the nine medical labels and the *Outside* label. We use categorical cross entropy as the objective

function. Similar to the CRF implementation, the Neural Net cost function also contains an L2-regularization component. We also use dropout (Srivastava et al., 2014) as an additional measure to avoid over-fitting. Fifty percent dropout is used to manipulate the inputs to the RNN and the Softmax layer. We use AdaGrad (Duchi et al., 2011) to optimize the network cost.

We use ten-fold cross validation to calculate the performance metric for each model. The dataset is divided at the note level. We separate out 10 % of the training set to form the validation set. This validation set is used to evaluate the different parameter combinations for CRF and RNN models. We employ early stopping to terminate the training run if the validation error increases consistently. We use a maximum of 40 epochs to train each network. The batch sizes used were kept constant at 128 for sentence level RNNs and 16 for document level RNNs.

We report micro-averaged recall, precision and f-score. We use exact phrase matching to calculate the evaluation score for our experiments. Each phrase labeled by the learned models is considered a true positive only if it matches the exact true boundary of the phrase and correctly labels all the words in the phrase.

We use CRFsuite (Okazaki, 2007) for implementing the CRF tagger. We use Lasagne to setup the Neural Net framework. Lasagne^2 is a machine learning library focused towards neural networks that is build on top of Theano (Bergstra et al., 2010).

6 Results

Table 2 shows the micro averaged scores for each method. All RNN models significantly outperform the baseline (CRF-context). Compared to the baseline system, our best system (GRU-document) improved the recall (0.8126), precision (0.7938) and F-score (0.8031) by 19%, 2% and 11 % respectively. Clearly the improvement in recall contributes more to the overall increase in system performance. The performance of different RNN models is almost similar, except for the GRU model which exhibits an F-score improvement of at least one percentage point over the rest.

Table 2

Cross validated micro-average of Precision, Recall and F-score for all medical tags

Models	Recall	Precision	F-score
CRF-nocontext	0.6562	0.7330	0.6925
CRF-context	0.6806	0.7711	0.7230
LSTM-sentence	0.8024	0.7803	0.7912
GRU-sentence	0.8013	0.7802	0.7906
LSTM-document	0.8050	0.7796	0.7921
GRU-document	**0.8126**	**0.7938**	**0.8031**

The changes (gain or loss) in label wise F-score for each RNN model relative to the baseline CRF-context method are plotted in Figure 2. GRU-document exhibits the highest gain overall in six of the nine tags: *indication* or diagnosis, *route*, *duration*, *severity*, *drug name*, and *other SSD*. For indication, its gain is about 0.19, a near 50% increase over the baseline. While the overall system performance of

GRU-sentence, LSTM-sentence and LSTM-document are very similar, they do exhibit somewhat varied performance for different labels. The sentence level models clearly outperform the document level RNNs (both GRU and LSTM) for *ADE* and *Dosage*. Additionally, GRU sentence model shows the highest gain in *ADE* f-score.
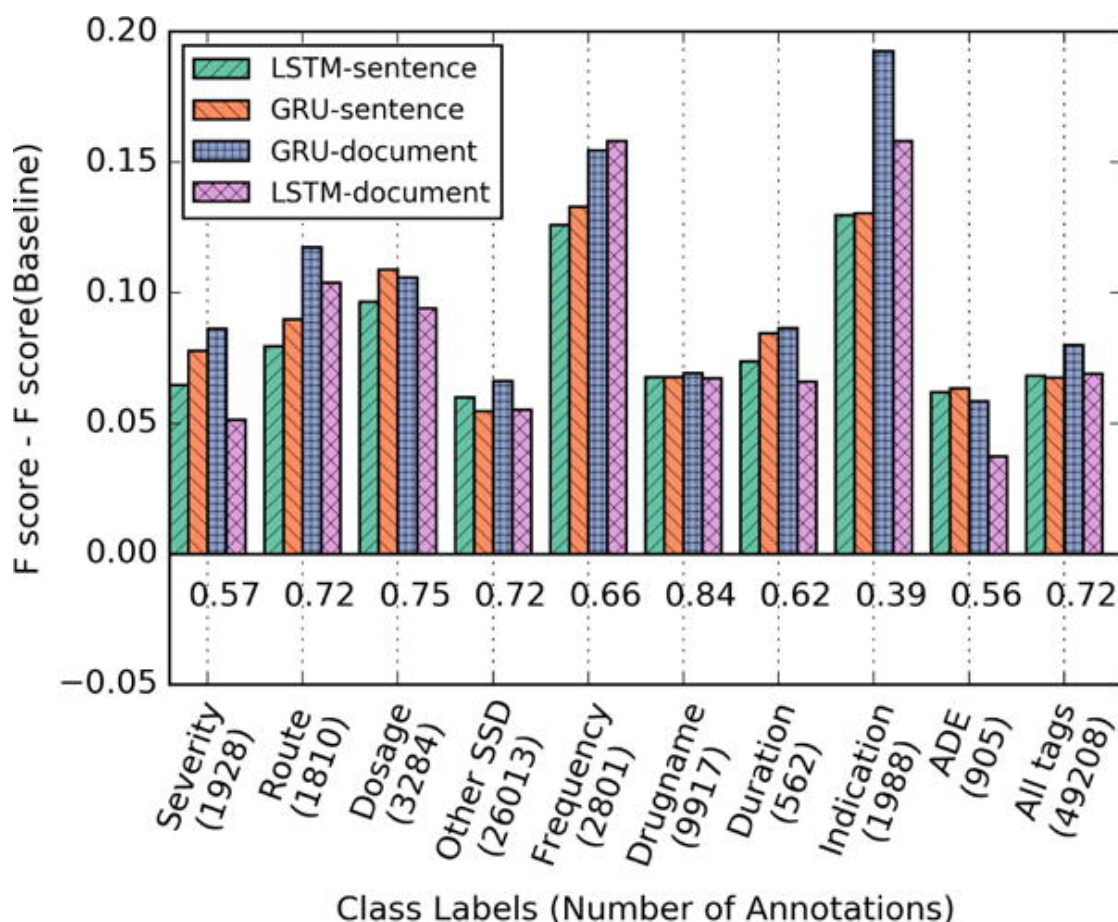

[Figure 2](#)

Change in F-score for RNN models with respect to CRF-context (baseline). The values below the plotted bars represent the baseline f-scores for each class label.

[Figure 3](#) shows the word level confusion matrix of different models for each label. Each cell shows the percentage of word tokens in row label *i* that were classified as column label *j*. The consistent increase of diagonal entries of RNN models for all ten labels, indicates an increase in the overall system accuracy when compared to the baseline. The most densely populated column in this figure is the *Outside* column, which denotes percentage of words that were erroneously labeled as *Outside*.
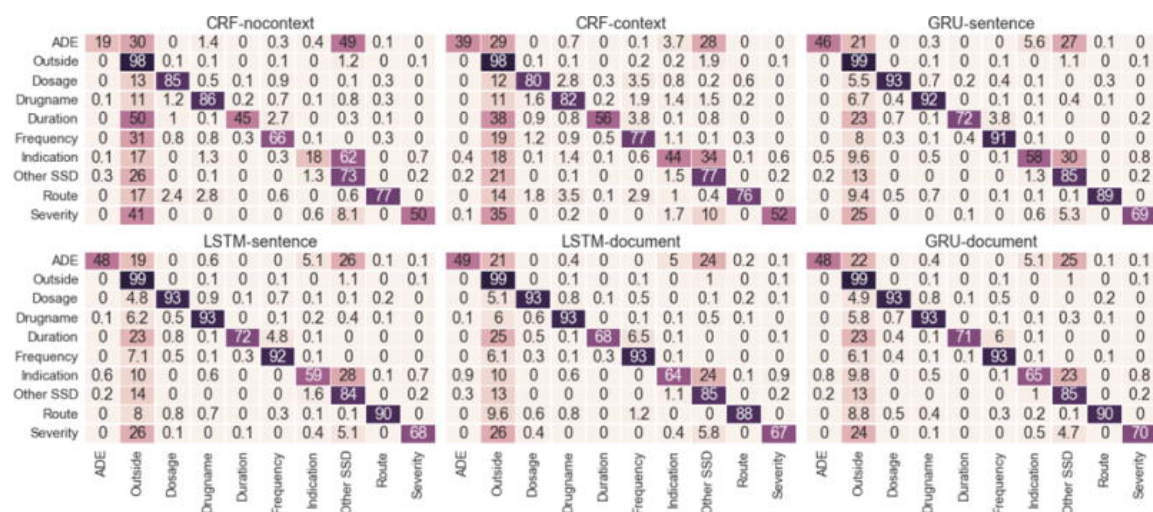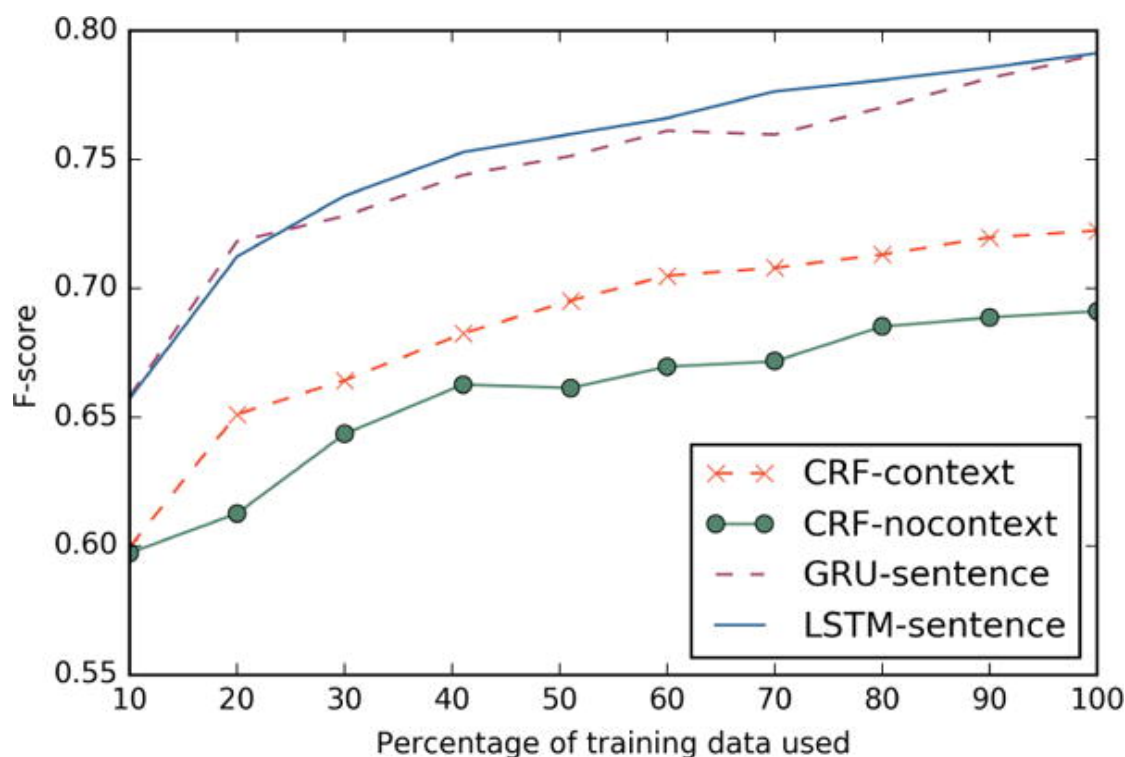
Figure 3

Heat-maps of Confusion Matrices of each method for the different class Labels. Rows are reference and columns are predictions. The value in cell (i, j) denotes the percentage of words in label i that were predicted as label j.

Figure 4 shows the change in average F-scores for each method with changing percentage of training data used. The setup for training, development and test data is kept the same as the ten-fold cross validation setup mentioned in Section 5. Only the training data is randomly down-sampled to achieve the reduced training data size. The figure shows that Recurrent Neural Network models perform better than traditional CRF models even with smaller training data sizes.

[Figure 4](#)

Change in F-score for all sentence models with respect to increasing training data size.

7 Discussion

We already discussed in the previous section how improved recall seems to be the major reason behind improvements in the RNN F-score. This trend can be observed in [Figure 3](#) where RNN models lead to significant decreases in confusion values present in *Outside* column.

Further examination of [Figure 3](#) shows two major sources of error in the CRF systems. The largest source of error is caused by confusing the relevant medical words as *Outside* (false negatives) and vice versa (false positives). The extent of false positives is not clear from [Figure 3](#), but can be estimated if one takes into account that even a 1 % confusion in the *Outside* row represents about 5000 words. The second largest source of error is the confusion among *ADE*, *Indication* and *Other SSD* labels. As we discuss in the following paragraphs, RNNs manage to significantly reduce both these type of errors.

The large improvement in recall of all labels for RNN models seems to suggest that RNNs are able to recognize a larger set of relevant patterns than CRF baselines. This supports our hypothesis that learning dependencies with variable context ranges is crucial for our task of medical information extraction from EHR notes. This is also evident from the reduced confusion among *ADE*, *Indication* and *Other SSD*. Since these tags share a common vocabulary of Sign, Symptom and Disease Names, identifying the underlying word or phrase is not enough to distinguish between the three. Use of relevant patterns from surrounding context is often needed as a discriminative cue. Consequently, *ADE*, *Indication* confusion values in the *Other SSD* column for RNNs exhibit significant decreases when compared to CRF-nocontext and CRF-context. We also see large improvements in detecting *Duration*, *Frequency* and *Severity*. The vocabulary of these labels often lack specific medical jargon terms. Examples of these labels include "seven days", "one week" for *duration*, "some", "small", "no

significant" for *severity* and "as needed", "twice daily" for *frequency*. Therefore, they are most likely to be confused with *Outside* label. This is indeed the case, as they have the highest confusion values in the *Outside* column of CRF-nocontext. Including context in CRF improves the performance, but not as much as RNN models which decrease the confusion by almost half or more in all cases. For example, GRU-document only confuses Frequency as an unlabeled word about 6.1 % of the time as opposed to 31 % and 19 % for CRF-nocontext and CRF-context respectively.

Document level models benefit by using context from outside the sentence. Since the label *Indication* requires the most use of surrounding context, it is clear that its performance would improve by using information from several sentences. Indications are diseases that are diagnosed by the medical staff, and the entire picture of the diagnosis is usually distributed across multiple sentences. Analysis of ADE is more complicated. Several ADE instances in a sentence also contain explicit cues similar to "secondary to" and "caused by". When coupled with *Drugnames* this is enough to classify the ADE. Sentence level models might depend more on these local cues which leads to improved performance. Document models, on the other hand, have to recognize patterns from a larger context, using a very small dataset (total ADE annotations are just 905) which is quite difficult.

The LSTM-document model does not show the same improvement over the sentence models as GRU-document. One possible reason for this might be the simpler recurrence structure of GRU neuron as compared to LSTM. Since there are only 780 document sequences in the dataset, the GRU model with a smaller number of trainable parameters might learn faster than LSTM. It is possible that with a larger dataset, LSTM might perform comparable to or better than GRU. However, our experiments with reducing the hidden layer size of LSTM-document model to control for the number of trainable parameters did not produce any significant improvements.

Moreover, figure 4 seems to indicate that there is not much difference between the performances of LSTM and GRU with different data sizes. However it is clearly surprising that RNN models with a larger number of parameters can still perform better than CRF models on smaller dataset sizes. This might be because the embedding layer, which contributes to a very large section of the trainable parameters, is initialized with a suitably good estimate using skip-gram word embeddings described in section 4.4.

8 Conclusion

We have shown that RNNs models like LSTM and GRU are valuable tools for extracting medical events and attributes from noisy natural language text of EHR notes. We believe that the significant improvement provided by gated RNN models is due to their ability to remember information across different range of dependencies as and when required. As mentioned previously in the introduction, this is very important for our task because different labels have different contextual dependencies. CRF models with hand crafted features like bag of words representation, use fixed context windows and lose a lot of information in the process.

RNNs are excellent in extracting relevant patterns from sequence data. However, they do not explicitly enforce constraints or dependencies over the output labels. We believe that adding a probabilistic graphical model framework for structured output prediction would further improve the performance of our system. This experiment remains as our future work.

Acknowledgments

Footnotes

[1]RNN Code is available at https://github.com/abhyudaynj/birnn-bionlp

[2]https://github.com/Lasagne/Lasagne

References

[Aronson2001] Aronson Alan R. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. Proceedings of the AMIA Symposium. 2001:17. [PMC free article] [PubMed] [Google Scholar]

[Bengio et al.1994] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks. 1994 Mar;5(2):157–166. [PubMed] [Google Scholar]

[Bergstra et al.2010] Bergstra James, Breuleux Olivier, Bastien Frdric, Lamblin Pascal, Pascanu Razvan, Desjardins Guillaume, Turian Joseph, Warde-Farley David, Bengio Yoshua. Theano: a CPU and GPU math expression compiler. Proceedings of the Python for Scientific Computing Conference (SciPy) 2010 [Google Scholar]

[Cho et al.2014] Cho Kyunghyun, van Merrienboer Bart, Bahdanau Dzmitry, Bengio Yoshua. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. ArXiv e-prints. 2014 Sep 1409:arXiv:1409.1259. [Google Scholar]

[Chung et al.2014] Chung Junyoung, Gulcehre Caglar, Cho KyungHyun, Bengio Yoshua. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv:1412.3555 [cs] 2014 Dec; arXiv: 1412.3555. [Google Scholar]

[Collier et al.2000] Collier NH, Nobata C, Tshjii J. Extracting the names of genes and gene products with a hidden markov model. Proceedings of the 18th International Conference on Computational Linguistics (COLING'2000) 2000:201–7. [Google Scholar]

[Collobert et al.2011] Collobert Ronan, Weston Jason, Bottou Lon, Karlen Michael, Kavukcuoglu Koray, Kuksa Pavel. Natural language processing (almost) from scratch. The Journal of Machine Learning Research. 2011;12:2493–2537. [Google Scholar]

[Da et al.1993] Lindberg Da, Humphreys Bl, McCray At. The Unified Medical Language System. Methods of information in medicine. 1993 Aug;32(4):281–291. [PMC free article] [PubMed] [Google Scholar]

[Duchi et al.2011] Duchi John, Hazan Elad, Singer Yoram. Adaptive subgradient methods for online learning and stochastic optimization. The Journal of Machine Learning Research. 2011;12:2121–2159. [Google Scholar]

[Friedman et al.1994] Friedman C, Alderson PO, Austin JH, Cimino JJ, Johnson SB. A general natural-language text processor for clinical radiology. Journal of the American Medical Informatics Association. 1994;1(2):161–174. [PMC free article] [PubMed] [Google Scholar]

[Gurulingappa et al.2010] Gurulingappa Harsha, Klinger Roman, Hofmann-Apitius Martin, Fluck Juliane. An Empirical Evaluation of Resources for the Identification of Diseases and Adverse Effects in Biomedical Literature. 2nd Workshop on Building and evaluating resources for biomedical text mining (7th edition of the Language Resources and Evaluation Conference) 2010 [Google Scholar]

[Haerian et al.2012] Haerian K, Salmasian H, Friedman C. Methods for Identifying Suicide or Suicidal Ideation in EHRs. AMIA Annual Symposium Proceedings. 2012 Nov;2012:1244–1253. [PMC free article] [PubMed] [Google Scholar]

[Hazlehurst et al.2005] Hazlehurst Brian, Frost H Robert, Sittig Dean F, Stevens Victor J. MediClass: A System for Detecting and Classifying Encounter-based Clinical Events in Any Electronic Medical Record. Journal of the American Medical Informatics Association: JAMIA. 2005;12(5):517–529. [PMC free article] [PubMed] [Google Scholar]

[Hirschman et al.2005] Hirschman L, Yeh A, Blaschke C, Valencia A. Overview of BioCreAtIvE: critical assessment of information extraction for biology. BMC Bioinformatics. 2005;6(Suppl 1):S1. [PMC free article] [PubMed] [Google Scholar]

[Hochreiter and Schmidhuber1997] Hochreiter Sepp, Schmidhuber Jrgen. Long Short-Term Memory. Neural Computation. 1997;9(8):1735–1780. [PubMed] [Google Scholar]

[Hochreiter et al.2001] Hochreiter Sepp, Bengio Yoshua, Frasconi Paolo, Schmidhuber Jrgen. Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies 2001 [Google Scholar]

[Huang et al.2015] Huang Zhiheng, Xu Wei, Yu Kai. Bidirectional LSTM-CRF Models for Sequence Tagging. arXiv:1508.01991 [cs] 2015 Aug; arXiv: 1508.01991. [Google Scholar]

[Jozefowicz et al.2015] Jozefowicz Rafal, Zaremba Wojciech, Sutskever Ilya. An Empirical Exploration of Recurrent Network Architectures. 2015:2342–2350. [Google Scholar]

[Kim et al.2009] Kim JD, Ohta T, Pyysalo S, Kano Y, Tsujii J. Overview of BioNLP'09 shared task on event extraction. Proceedings of the Workshop on BioNLP: Shared Task. 2009:1–9. [Google Scholar]

[Lafferty et al.2001] Lafferty John D, McCallum Andrew, Pereira Fernando CN. Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc; 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data; pp. 282–289. [Google Scholar]

[Li et al.2009] Li Z, Cao Y, Antieau L, Agarwal S, Zhang Q, Yu H. Extracting Medication Information from Patient Discharge Summaries. Third i2b2 Shared-Task Workshop 2009 [Google Scholar]

[Li et al.2010] Li Z, Liu F, Antieau L, Cao Y, Yu H. Lancet: a high precision medication event extraction system for clinical text. Journal of the American Medical Informatics Association. 2010;17(5):563–567. [PMC free article] [PubMed] [Google Scholar]

[McCallum et al.2000] McCallum Andrew, Freitag Dayne, Pereira Fernando CN. Proceedings of the Seventeenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc; 2000. Maximum Entropy Markov Models for Information Extraction and Segmentation; pp. 591–598. [Google Scholar]

[Mikolov et al.2013] Mikolov Tomas, Sutskever Ilya, Chen Kai, Corrado Greg S, Dean Jeff. Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems. 2013:3111–3119. [Google Scholar]

[Okazaki2007] Okazaki Naoaki. CRFsuite: a fast implementation of Conditional Random Fields (CRFs) 2007 [Google Scholar]

[Polepalli Ramesh et al.2014] Ramesh Balaji Polepalli, Belknap Steven M, Li Zuofeng, Frid Nadya, West Dennis P, Yu Hong. Automatically Recognizing Medication and Adverse Event Information From Food and Drug Administrations Adverse Event Reporting System Narratives. JMIR Medical Informatics. 2014 Jun;2(1):e10. 00001. [PMC free article] [PubMed] [Google Scholar]

[Pradhan et al.2014] Pradhan Sameer, Elhadad Nomie, South Brett R, Martinez David, Christensen Lee, Vogel Amy, Suominen Hanna, Chapman Wendy W, Savova Guergana. Evaluating the state of the art in disorder recognition and normalization of the clinical narrative. Journal of the American Medical Informatics Association: JAMIA. 2014 Aug; 00002. [PMC free article] [PubMed] [Google Scholar]

[Rochefort et al.2015] Rochefort Christian M, Verma Aman D, Eguale Tewodros, Lee Todd C, Buckeridge David L. A novel method of adverse event detection can accurately identify venous thromboembolisms (VTEs) from narrative electronic health record data. Journal of the American Medical Informatics Association: JAMIA. 2015 Jan;22(1):155–165. [PMC free article] [PubMed] [Google Scholar]

[Srivastava et al.2014] Srivastava Nitish, Hinton Geoffrey, Krizhevsky Alex, Sutskever Ilya, Salakhutdinov Ruslan. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. J Mach Learn Res. 2014 Jan15(1):1929–1958. [Google Scholar]

[Tang et al.2013] Tang Buzhou, Wu Yonghui, Jiang Min, Denny Joshua C, Xu Hua. Recognizing and Encoding Disorder Concepts in Clinical Text using Machine Learning and Vector Space Model. 2013 00004. [Google Scholar]

[Wang et al.2015] Wang Chang, Cao Liangliang, Zhou Bowen. Medical Synonym Extraction with Concept Space Models. arXiv:1506.00528 [cs] 2015 Jun; arXiv: 1506.00528. [Google Scholar]

[Xu et al.2010] Xu H, Stenner SP, Doan S, Johnson KB, Waitman LR, Denny JC. MedEx: a medication information extraction system for clinical narratives. J Am Med Inform Assoc. 2010;17:19–24. 1. [PMC free article] [PubMed] [Google Scholar]