Clinical Information Extraction via Convolutional Neural Network

Abstract

We report an implementation of a clinical information extraction tool that leverages deep neural network to annotate event spans and their attributes from raw clinical notes and pathology reports. Our approach uses context words and their part-of-speech tags and shape information as features. Then we hire temporal (1D) convolutional neural network to learn hidden feature representations. Finally, we use Multilayer Perceptron (MLP) to predict event spans. The empirical evaluation demonstrates that our approach significantly outperforms baselines.

Introduction

In the past few years, there has been much interest in applying neural network based deep learning techniques to solve all kinds of natural language processing (NLP) tasks. From low level tasks such as language modeling, POS tagging, named entity recognition, and semantic role labeling BIBREF0 , BIBREF1 , to high level tasks such as machine translation, information retrieval, semantic analysis BIBREF2 , BIBREF3 , BIBREF4 and sentence relation modeling tasks such as paraphrase identification and question answering BIBREF5 , BIBREF6 , BIBREF7 . Deep representation learning has demonstrated its importance for these tasks. All the tasks get performance improvement via learning either word level representations or sentence level representations.

In this work, we brought deep representation learning technologies to the clinical domain. Specifically, we focus on clinical information extraction, using clinical notes and pathology reports from the Mayo Clinic. Our system will identify event expressions consisting of the following components:

The input of our system consists of raw clinical notes or pathology reports like below:

And output annotations over the text that capture the key information such as event mentions and attributes. Table TABREF7 illustrates the output of clinical information extraction in details.

To solve this task, the major challenge is how to precisely identify the spans (character offsets) of the event expressions from raw clinical notes. Traditional machine learning approaches usually build a supervised classifier with features generated by the Apache clinical Text Analysis and Knowledge Extraction System (cTAKES) . For example, BluLab system BIBREF8 extracted morphological(lemma), lexical(token), and syntactic(part-of-speech) features encoded from cTAKES. Although using the domain specific information extraction tools can improve the performance, learning how to use it well for clinical domain feature engineering is still very time-consuming. In short, a simple and effective method that only leverage basic NLP modules and achieves high extraction performance is desired to save costs.

To address this challenge, we propose a deep neural networks based method, especially convolution neural network BIBREF0 , to learn hidden feature representations directly from raw clinical notes. More specifically, one method first extract a window of surrounding words for the candidate word. Then, we attach each word with their part-of-speech tag and shape information as extra features. Then our system deploys a temporal convolution neural network to learn hidden feature representations. Finally, our system uses Multilayer Perceptron (MLP) to predict event spans. Note that we use the same model to predict event attributes.

Constructing High Quality Training Dataset

The major advantage of our system is that we only leverage NLTK tokenization and a POS tagger to preprocess our training dataset. When implementing our neural network based clinical information

extraction system, we found it is not easy to construct high quality training data due to the noisy format of clinical notes. Choosing the proper tokenizer is quite important for span identification. After several experiments, we found "RegexpTokenizer" can match our needs. This tokenizer can generate spans for each token via sophisticated regular expression like below,

Neural Network Classifier

Event span identification is the task of extracting character offsets of the expression in raw clinical notes. This subtask is quite important due to the fact that the event span identification accuracy will affect the accuracy of attribute identification. We first run our neural network classifier to identify event spans. Then, given each span, our system tries to identify attribute values.

Temporal Convolutional Neural Network

The way we use temporal convlution neural network for event span and attribute classification is similar with the approach proposed by BIBREF0 . Generally speaking, we can consider a word as represented by INLINEFORM0 discrete features INLINEFORM1 , where INLINEFORM2 is the dictionary for the INLINEFORM3 feature. In our scenario, we just use three features such as token mention, pos tag and word shape. Note that word shape features are used to represent the abstract letter pattern of the word by mapping lower-case letters to “x”, upper-case to “X”, numbers to “d”, and retaining punctuation. We associate to each feature a lookup table. Given a word, a feature vector is then obtained by concatenating all lookup table outputs. Then a clinical snippet is transformed into a word embedding matrix. The matrix can be fed to further 1-dimension convolutional neural network and max pooling layers. Below we will briefly introduce core concepts of Convoluational Neural Network (CNN).

Temporal Convolution applies one-dimensional convolution over the input sequence. The

one-dimensional convolution is an operation between a vector of weights INLINEFORM0 and a vector of inputs viewed as a sequence INLINEFORM1 . The vector INLINEFORM2 is the filter of the convolution. Concretely, we think of INLINEFORM3 as the input sentence and INLINEFORM4 as a single feature value associated with the INLINEFORM5 -th word in the sentence. The idea behind the one-dimensional convolution is to take the dot product of the vector INLINEFORM6 with each INLINEFORM7 -gram in the sentence INLINEFORM8 to obtain another sequence INLINEFORM9 : DISPLAYFORM0

Usually, INLINEFORM0 is not a single value, but a INLINEFORM1 -dimensional word vector so that INLINEFORM2 . There exist two types of 1d convolution operations. One was introduced by BIBREF9 and also known as Time Delay Neural Networks (TDNNs). The other one was introduced by BIBREF0 . In TDNN, weights INLINEFORM3 form a matrix. Each row of INLINEFORM4 is convolved with the corresponding row of INLINEFORM5 . In BIBREF0 architecture, a sequence of length INLINEFORM6 is represented as: DISPLAYFORM0

where INLINEFORM0 is the concatenation operation. In general, let INLINEFORM1 refer to the concatenation of words INLINEFORM2 . A convolution operation involves a filter INLINEFORM3 , which is applied to a window of INLINEFORM4 words to produce the new feature. For example, a feature INLINEFORM5 is generated from a window of words INLINEFORM6 by: DISPLAYFORM0

where INLINEFORM0 is a bias term and INLINEFORM1 is a non-linear function such as the hyperbolic tangent. This filter is applied to each possible window of words in the sequence INLINEFORM2 to produce the feature map: DISPLAYFORM0

where INLINEFORM0 .

We also employ dropout on the penultimate layer with a constraint on INLINEFORM0 -norms of the

weight vector. Dropout prevents co-adaptation of hidden units by randomly dropping out a proportion INLINEFORM1 of the hidden units during forward-backpropagation. That is, given the penultimate layer INLINEFORM2 , instead of using: DISPLAYFORM0

for output unit INLINEFORM0 in forward propagation, dropout uses: DISPLAYFORM0

where INLINEFORM0 is the element-wise multiplication operator and INLINEFORM1 is a masking vector of Bernoulli random variables with probability INLINEFORM2 of being 1. Gradients are backpropagated only through the unmasked units. At test step, the learned weight vectors are scaled by INLINEFORM3 such that INLINEFORM4 , and INLINEFORM5 is used to score unseen sentences. We additionally constrain INLINEFORM6 -norms of the weight vectors by re-scaling INLINEFORM7 to have INLINEFORM8 whenever INLINEFORM9 after a gradient descent step.

Dataset

We use the Clinical TempEval corpus as the evaluation dataset. This corpus was based on a set of 600 clinical notes and pathology reports from cancer patients at the Mayo Clinic. These notes were manually de-identified by the Mayo Clinic to replace names, locations, etc. with generic placeholders, but time expression were not altered. The notes were then manually annotated with times, events and temporal relations in clinical notes. These annotations include time expression types, event attributes and an increased focus on temporal relations. The event, time and temporal relation annotations were distributed separately from the text using the Anafora standoff format. Table TABREF19 shows the number of documents, event expressions in the training, development and testing portions of the 2016 THYME data.

Evaluation Metrics

All of the tasks were evaluated using the standard metrics of precision(P), recall(R) and INLINEFORM0 : DISPLAYFORM0

where INLINEFORM0 is the set of items predicted by the system and INLINEFORM1 is the set of items manually annotated by the humans. Applying these metrics of the tasks only requires a definition of what is considered an "item" for each task. For evaluating the spans of event expressions, items were tuples of character offsets. Thus, system only received credit for identifying events with exactly the same character offsets as the manually annotated ones. For evaluating the attributes of event expression types, items were tuples of (begin, end, value) where begin and end are character offsets and value is the value that was given to the relevant attribute. Thus, systems only received credit for an event attribute if they both found an event with correct character offsets and then assigned the correct value for that attribute BIBREF10 .

Hyperparameters and Training Details

We want to maximize the likelihood of the correct class. This is equivalent to minimizing the negative log-likelihood (NLL). More specifically, the label INLINEFORM0 given the inputs INLINEFORM1 is predicted by a softmax classifier that takes the hidden state INLINEFORM2 as input: DISPLAYFORM0

After that, the objective function is the negative log-likelihood of the true class labels INLINEFORM0 : DISPLAYFORM0

where INLINEFORM0 is the number of training examples and the superscript INLINEFORM1 indicates the INLINEFORM2 th example.

We use Lasagne deep learning framework. We first initialize our word representations using publicly

available 300-dimensional Glove word vectors . We deploy CNN model with kernel width of 2, a filter size of 300, sequence length is INLINEFORM0 , number filters is INLINEFORM1 , stride is 1, pool size is INLINEFORM2 , cnn activation function is tangent, MLP activation function is sigmoid. MLP hidden dimension is 50. We initialize CNN weights using a uniform distribution. Finally, by stacking a softmax function on top, we can get normalized log-probabilities. Training is done through stochastic gradient descent over shuffled mini-batches with the AdaGrad update rule BIBREF11 . The learning rate is set to 0.05. The mini-batch size is 100. The model parameters were regularized with a per-minibatch L2 regularization strength of INLINEFORM3 .

Results and Discussions

Table TABREF28 shows results on the event expression tasks. Our initial submits RUN 4 and 5 outperformed the memorization baseline on every metric on every task. The precision of event span identification is close to the max report. However, our system got lower recall. One of the main reason is that our training objective function is accuracy-oriented. Table TABREF29 shows results on the phase 2 subtask.

Conclusions

In this paper, we introduced a new clinical information extraction system that only leverage deep neural networks to identify event spans and their attributes from raw clinical notes. We trained deep neural networks based classifiers to extract clinical event spans. Our method attached each word to their part-of-speech tag and shape information as extra features. We then hire temporal convolution neural network to learn hidden feature representations. The entire experimental results demonstrate that our approach consistently outperforms the existing baseline methods on standard evaluation datasets.

Our research proved that we can get competitive results without the help of a domain specific feature extraction toolkit, such as cTAKES. Also we only leverage basic natural language processing modules such as tokenization and part-of-speech tagging. With the help of deep representation learning, we can dramatically reduce the cost of clinical information extraction system development.