NLU Assignment 3

Rishi Hazra
Systems Engineering
14542
rishixtreme@gmail.com

Abstract

Named-entity recognition (NER) is an information extraction process that seeks to locate and classify named entities in text into pre-defined categories such as the names of persons, organizations, locations, expressions of times etc. Medical entities can be diseases, drugs, symptoms, etc. Previously, researchers in the field have used hand crafted features to identify medical entities in medical literature. It has been found that in contrast with semantic approaches which require rich domain knowledge for rule or pattern construction, statistical approaches are more scalable.

1 Introduction

Named entity recognition (NER) is a challenging learning problem. On one hand, in most languages and domains, there is only a very small amount of supervised training data available. On the other hand, the are few constraints on the kinds of words that can be names, so generalizing from this small sample of data is difcult. That is why, tagging decisions for each token is important. We compare two models here (i) conditional random field and (ii) a bidirectional LSTM with a sequential conditional random layer above it.

1.1 Dataset

A training dataset of labeled sentences is given. The format of each line in the training dataset is token label. There is one token per line followed by a space and its label. Blank lines indicate the end of a sentence. It has a total of 3655 sentences.

1.2 Problem Statement

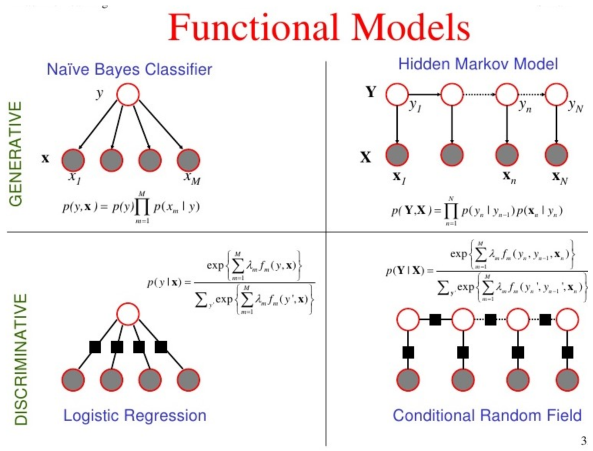
The goal of the assignment is to build an NER system for diseases and treatments. The input of the code will be a set of tokenized sentences and the output will be a label for each token in the sentence. Labels can be D, T or O signifying disease, treatment or other. We need to write a sequence tagger that labels the given sentences in a tokenized test file. The tokenized test file follows the same format as training except that it does not have the final label in the input. The output should label the test file in the same format as the training data.

2 Approach

Bi-LSTM and CRF models were implemented in this assignment. Pycrfsuite was used to implement the Conditional Random Field. 'Keras' was used for implementing Bi-LSTM with CRF.

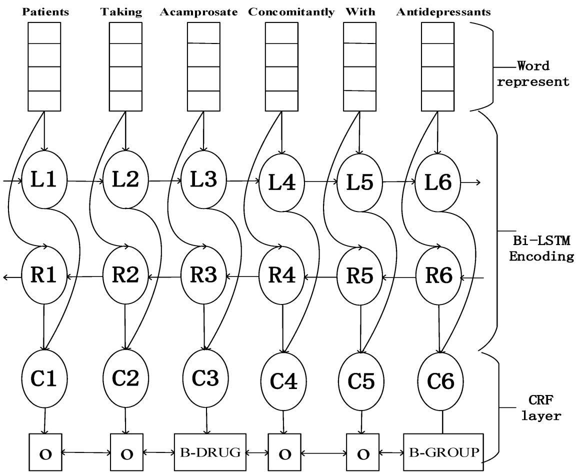
2.1 Conditional Random Field

Conditional random fields (CRFs) are a class of statistical modeling method applied in machine learning and used for structured prediction. They are used for discriminative undirected probabilistic graphical models. It is often used for labeling and parsing of sequential data. Specific applications include named entity recognition (as an alternative to HMM).


Functional Models
Naïve Bayes Classifier
Hidden Markov Model
GENERATIVE
DISCRIMINATIVE
y
x
x_1
x_M
$p(y,\mathbf{x}) = p(y)\prod_{m=1}^{M} p(x_m \mid y)$
Y
X
y_1
y_n
y_N
$\mathbf{x}_1$
$\mathbf{x}_n$
$\mathbf{x}_N$
$p(\mathbf{Y},\mathbf{X}) = \prod_{n=1}^{N} p(y_n \mid y_{n-1}) p(\mathbf{x}_n \mid y_n)$
$p(y \mid \mathbf{x}) = \frac{\exp\left\{\sum_{m=1}^{M} \lambda_m f_m(y,\mathbf{x})\right\}}{\sum_{y'} \exp\left\{\sum_{m=1}^{M} \lambda_m f_m(y',\mathbf{x})\right\}}$
$p(\mathbf{Y} \mid \mathbf{X}) = \frac{\exp\left\{\sum_{m=1}^{M} \lambda_m f_m(y_n, y_{n-1}, \mathbf{x}_n)\right\}}{\sum_{y'} \exp\left\{\sum_{m=1}^{M} \lambda_m f_m(y_n', y_{n-1}', \mathbf{x}_n)\right\}}$
Logistic Regression
Conditional Random Field
3

2.2 Bi-LSTM with CRF

For a named entity recognition task, neural network based methods are very popular and common. In order to capture the sequence of our data, we use the LSTM model with a sequential conditional random layer above it. Encoded input is fed to the Bi-LSTM layer and the CRF layer gives a probability distribution as its output which is then used to predict the output tag.



2.3 Metric used

We use precision and recall as the metric for comparing our models since the $\langle O \rangle$ dominates other labels. In other words, the label distribution is highly skewed.

$$precision = \frac{t_p}{t_p + f_p}$$

$$recall = \frac{t_p}{t_p + f_n}$$

t_p: number of true positives; f_p: number of false positives; f_n: number of false negatives;

3 Procedure

For Bi-LSTM, max length of sentences are fixed at 100 and short sentences are zero padded (zero represents $\langle UNK \rangle$ token). The encoded input is split into batches of size 25 and is given as input to an embedding layer which is followed by a Bi-LSTM layer of size 50 and dropout 0.1. The output is fed to a CRF layer. the process is run for 5 epochs with a validation split of 0.05.

Feature	Accuracy	Precision	Recall
Word tagger	0.89	0.88	0.89
Embedding tagger	0.92	0.92	0.92
POS Tagger	0.90	0.89	0.90

3.1 Confusion Matrix for Bi-LSTM CRF model

	O	D	T
O	24396	129	49
D	347	511	7
T	153	33	157

The Bi-LSTM CRF model is better than the pycrfsuite model. Also, adding an embedding layer before feeding the input into LSTM layer improves the precision and recall by approximately 25%.