Bidirectional LSTM-CRF Models for Sequence Tagging

Name : Archana Kalburgi

Course : CS 541-B Artificial Intelligence

Email ID : akalburg@stevens.edu

Table of Contents

Abstract

Named Entity Recognition (NER) is a problem in Natural Language Processing (NLP), which involves locating and classifying names (people, places, organizations, etc.) that appear in unstructured text. Many of the NLP applications that use this problem are dealing with use-cases such as machine translation, information retrieval, chatbots, etc. In a nutshell, for each training sentence, I would like to predict what "tag" each token will have.

In this paper the authors have proposed a variety of **Long Short-Term Memory (LSTM)** based models for sequence tagging. These models include **LSTM networks**, **bidirectional LSTM (BI-LSTM)** networks, **LSTM with a Conditional Random Field (CRF)** layer (LSTM-CRF) and **bidirectional LSTM with a CRF layer (BI-LSTM-CRF)**.

The tagging is achieved by applying a bidirectional LSTM CRF (denoted as BI-LSTM-CRF) model to **NLP benchmark sequence tagging data sets**. The **bidirectional LSTM** component enables the BILSTM-CRF model to efficiently use both past and future input features. And the **CRF layer** enables it to use sentence level tag information. The BI-LSTMCRF model can produce state of the art accuracy of 97.55% on POS, 94.46% on chunking and 90.10% on NER data sets. In addition, it is robust and has less dependency on word embedding as compared to previous observations.

The progress of the project and the source code can be tracked within this **git repository**.

Dataset details

The dataset used in this project is an annotated GMB(Groningen Meaning Bank) corpus for entity classification with enhanced and popular features by Natural Language Processing applied to the data set which is built specifically to train the classifier to predict named entities such as name, location, etc.

Tags

The tags used in datasets are in the form of "B-geo", "-org", "O", the **I-prefix** indicates that the tag is inside a chunk (i.e. a noun group, a verb group etc.); the **O-prefix** indicates that the token belongs to no chunk; the **B-prefix** indicates that the tag is at the beginning of a chunk that follows another chunk without O tags between the two chunks

Statistics

The dataset has 1354149 number of words and the target column is titled “tag”.

Tagged entities

The number of tagged entities are as detailed below:
'O': 1146068', geo-nam': 58388, 'org-nam': 48034, 'per-nam': 23790, 'gpe-nam': 20680, 'tim-dat': 12786, 'tim-dow': 11404, 'per-tit': 9800, 'per-fam': 8152, 'tim-yoc': 5290, 'tim-moy': 4262, 'per-giv': 2413, 'tim-clo': 891, 'art-nam': 866, 'eve-nam': 602, 'nat-nam': 300, 'tim-nam': 146, 'eve-ord': 107, 'per-ini': 60, 'org-leg': 60, 'per-ord': 38, 'tim-dom': 10, 'per-mid': 1, 'art-add': 1

The entity tag used in this dataset is as follows:

Tag	Meaning	Example
geo	Geography	Britain

org	Organization	IAEA
per	Person	Thomas
gpe	Geopolitical entity	Indian
tim	Time	Wednesday
art	Artifact	Pentastar
eve	Event	Armistice
nat	Natural phenomenon	H5N1

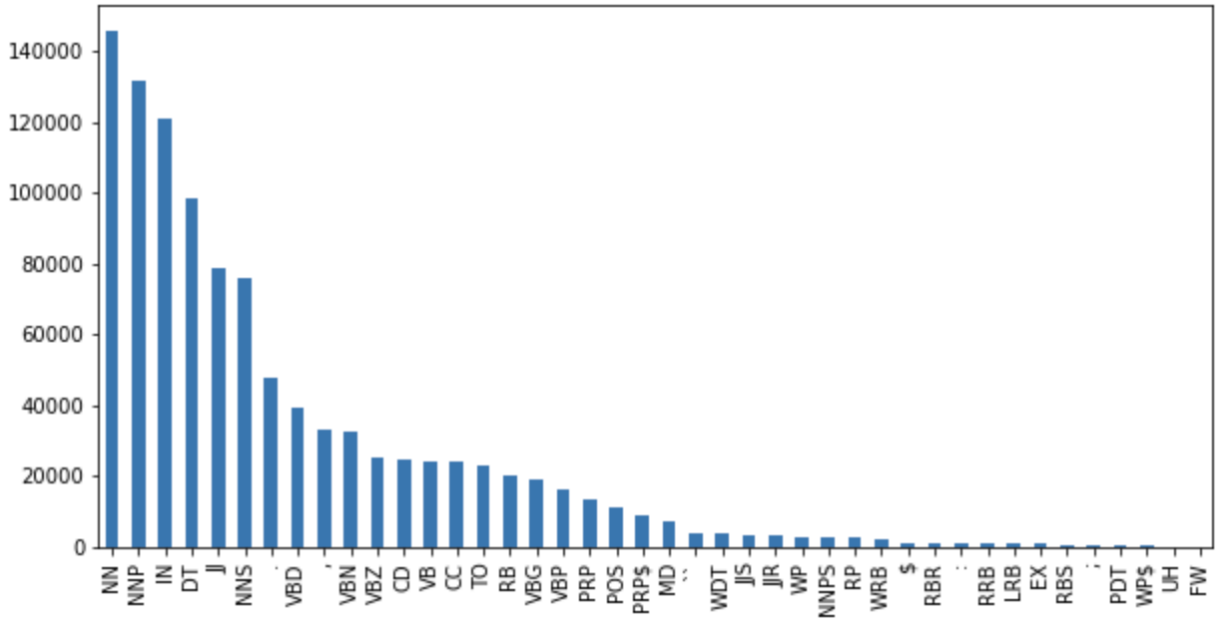
Data preprocessing and visualization

Majority of the data was clean. As a part of data preprocessing I have replaced the missing values with the last valid observation by propagating it forward to the next valid backfill. Following screenshots depict the difference between before and after replacing null values.

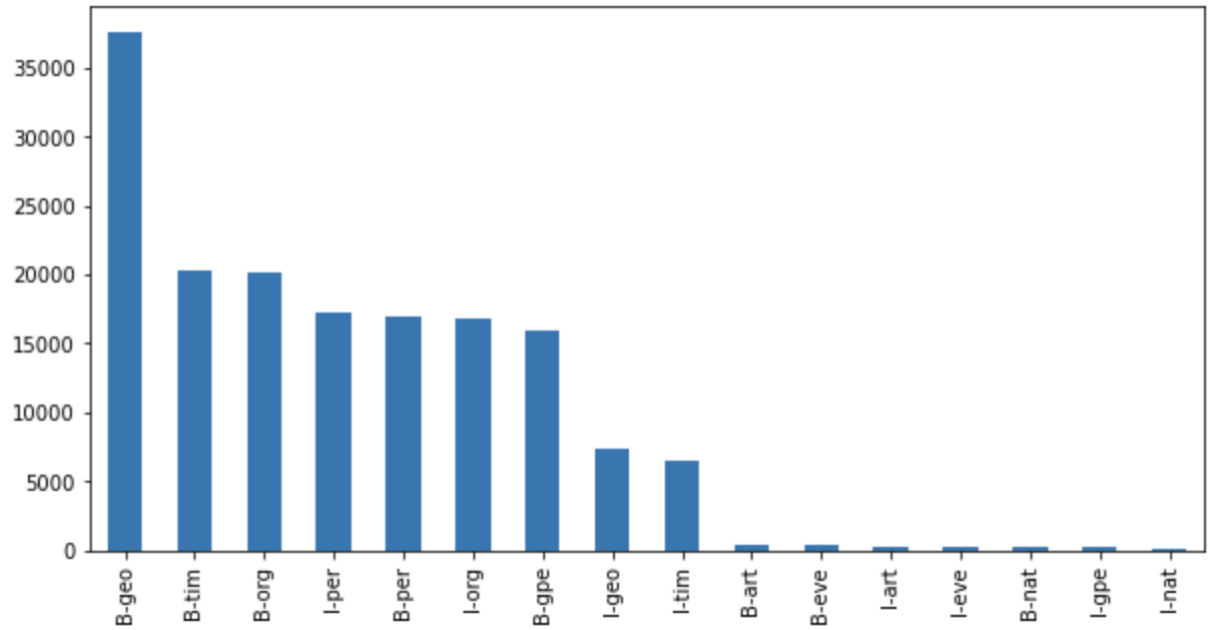
	Sentence #	Word	POS	Tag
0	Sentence: 1	Thousands	NNS	O
1	NaN	of	IN	O
2	NaN	demonstrators	NNS	O
3	NaN	have	VBP	O
4	NaN	marched	VBN	O

	Sentence #	Word	POS	Tag
0	1	Thousands	NNS	O
1	1	of	IN	O
2	1	demonstrators	NNS	O
3	1	have	VBP	O
4	1	marched	VBN	O

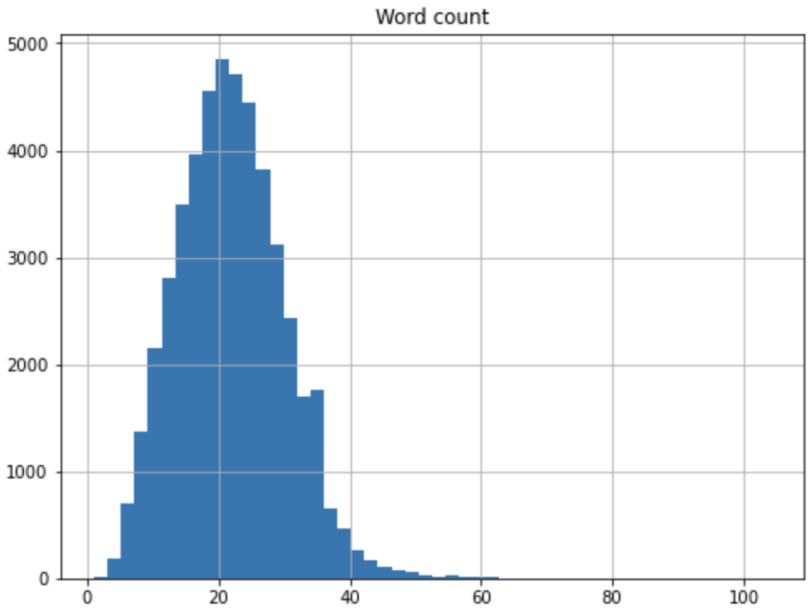
In the dataset there are 42 unique POS tags and 17 unique NER tags. The following plots depicts the distribution of the tags



Distribution of POS tags



Distribution of NER tags



Distribution of the words

Train, test and validation sets

The following table shows the size of the sentences, tokens, and labels for training, validation and test sets respectively.

Table 1: Size of sentences, tokens, and labels for training, validation and test sets.

		POS	CoNLL2000	CoNLL2003
training	sentence #	39831	8936	14987
	token #	950011	211727	204567
validation	sentence #	1699	N/A	3466
	token #	40068	N/A	51578
test	sentences #	2415	2012	3684
	token #	56671	47377	46666
	label #	45	22	9

More details about the data can be found here.

Tools and technologies

Software/ library packages

This project is implemented using python 3.9.1. And I will be making use of the following library package.

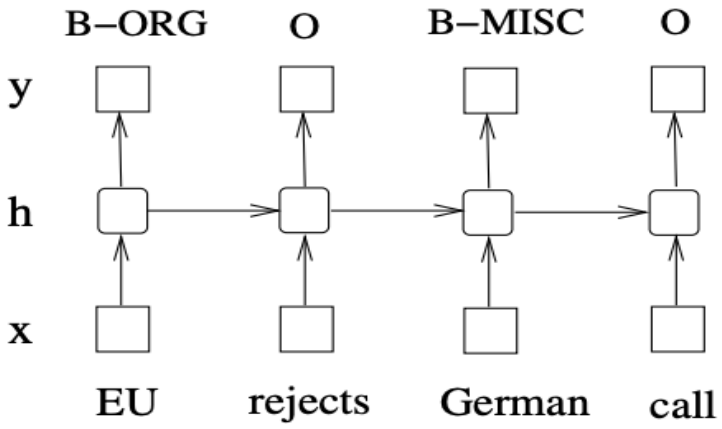
1. Keras 2.2.4
2. Pickle
3. Regular expression: re
4. Pandas library
5. Numpy library
6. Matplot library for visualization
7. Scikit library
8. Tensorflow

Hardware setup

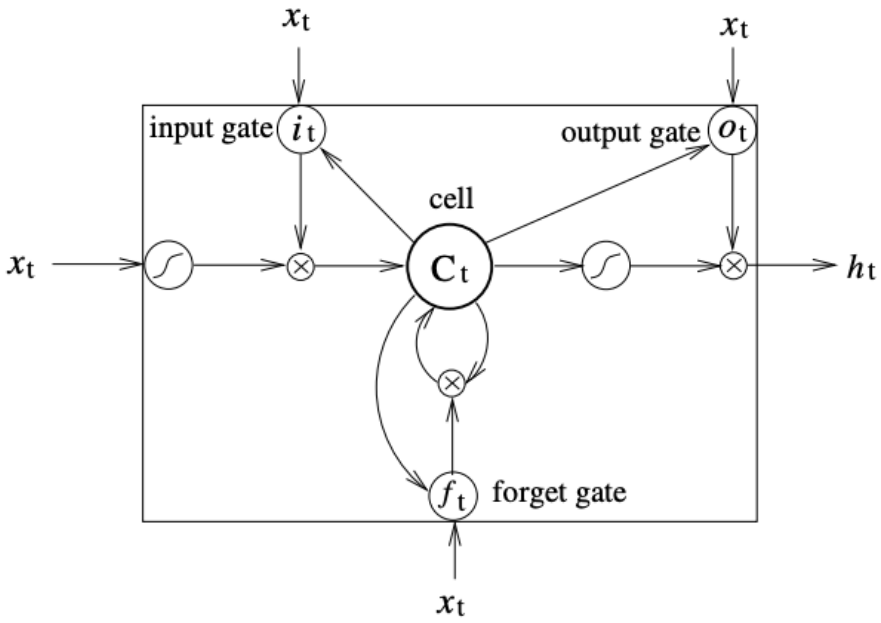
In this project I am using GPU 1.13.1. The models will be run on my local machine where the data to train the model are stored.

Experiments

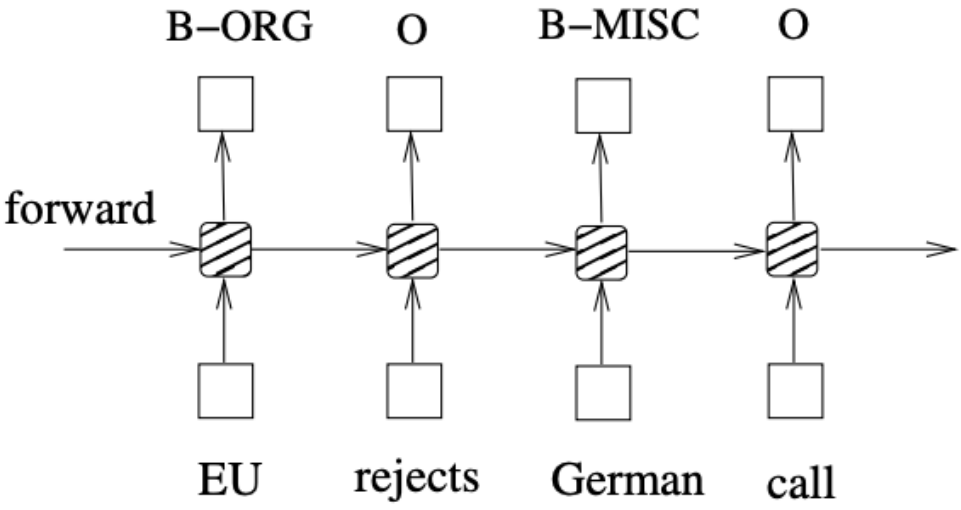
Neural network architecture



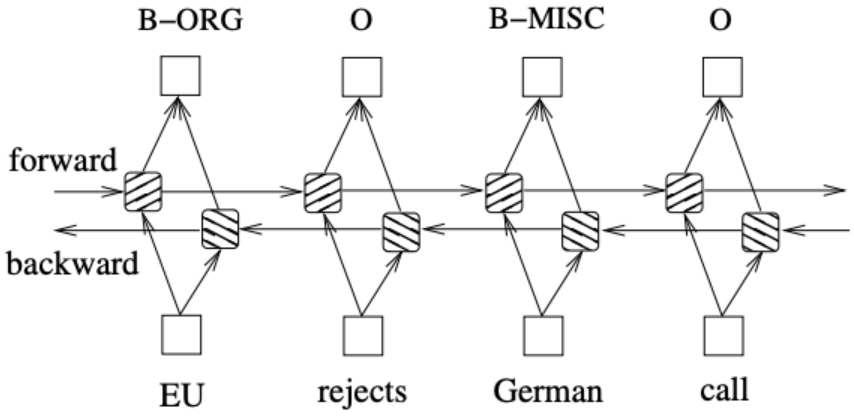
A simple RNN network



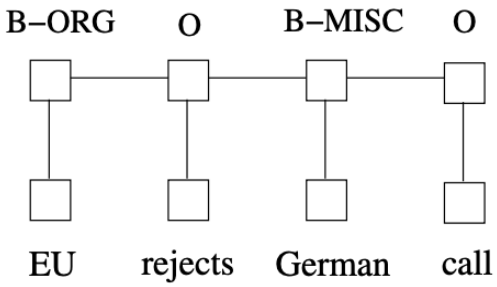
Long-short term memory cell



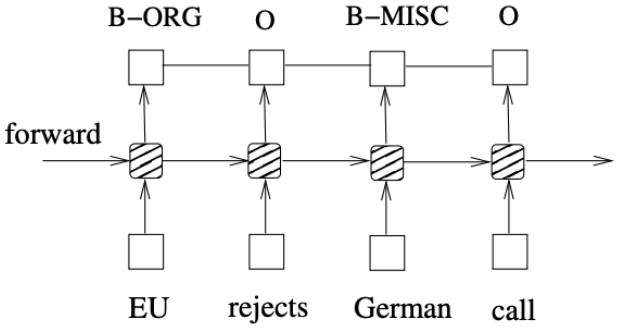
A LSTM network



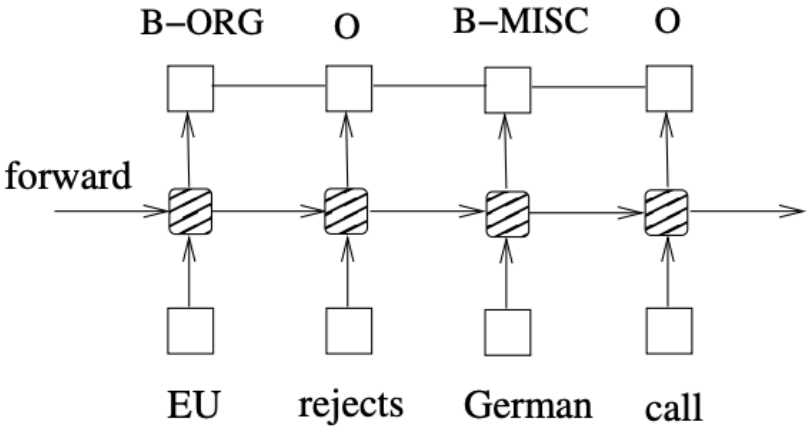
Bidirectional LSTM network



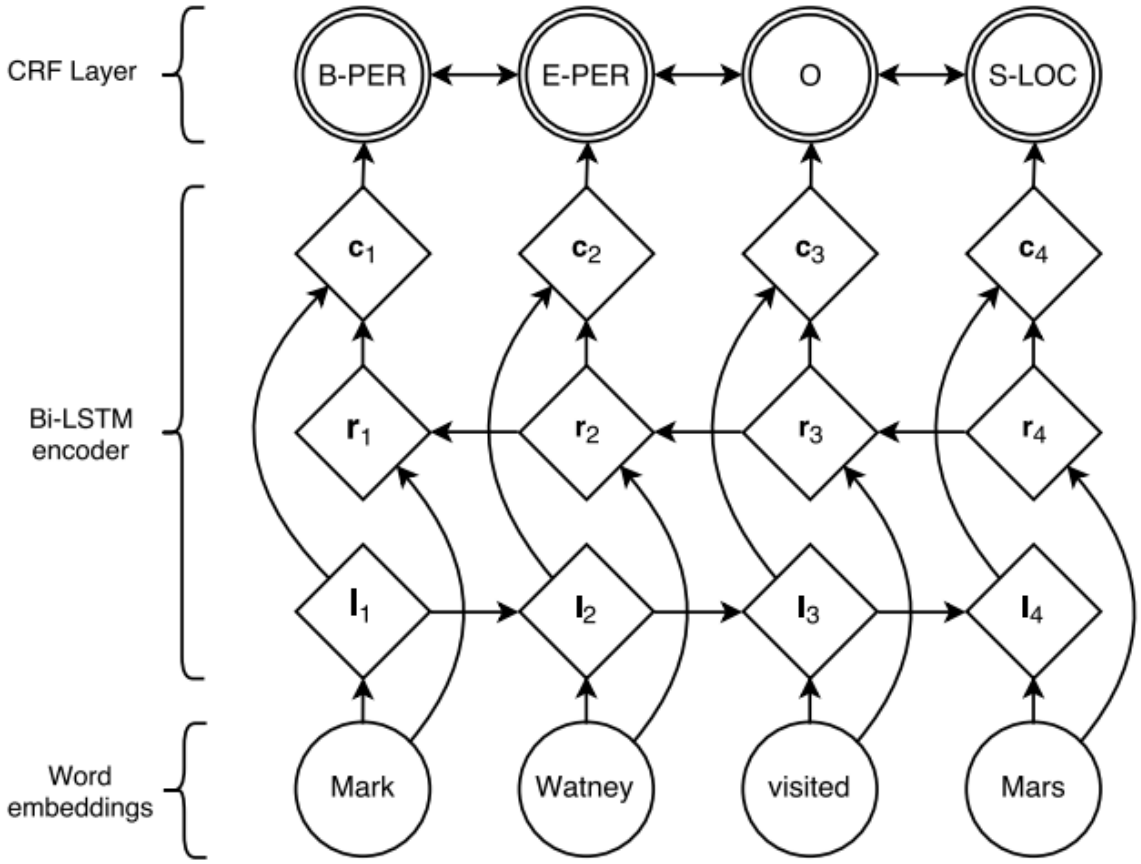
A CRF network



A LSTM-CRF model



A BI-LSTM-CRF model.



Main architecture of the network. Word embeddings are given to a bidirectional LSTM. *li* represents the word *i* and its left context, *ri* represents the word *i* and its right context. Concatenating these two vectors yields a representation of the word i in its context, *ci.*