Name Entity Recognition

Gao Xin
Wang Han
Wu Xin Yun
Xiong Mai He

01

Introduction

NER: A recipe for unstructured data

Our Data

47,959 **sentences** &

35,178 **words**

Background Knowledge

	Sentence #	Word	POS	Tag
0	Sentence: 1	Thousands	NNS	O
1	Sentence: 1	of	IN	O
2	Sentence: 1	demonstrators	NNS	O
3	Sentence: 1	have	VBP	O
4	Sentence: 1	marched	VBN	O
5	Sentence: 1	through	IN	O
6	Sentence: 1	London	NNP	B-geo
7	Sentence: 1	to	TO	O
8	Sentence: 1	protest	VB	O
9	Sentence: 1	the	DT	O

BIO & POS

Tag	Label meaning	Example Given
geo	Geographical Entity	London
org	Organization	ONU
per	Person	Bush
gpe	Geopolitical Entity	British
tim	Time indicator	Wednesday
art	Artifact	Chrysler
eve	Event	Christmas
nat	Natural Phenomenon	Hurricane
O	No-Label	the

Lexical Term	Tag	Example
Noun	NN	Paris, France, Someone
Verb	VB	Work, train, learn
Determiner	DT	The, a

Preprocessing

Techniques

Create a Vocabulary:

Word2idx: This dictionary has all the **unique words**(terms) as keys with a corresponding unique **ID** as values

Tag2idx: This is the **reverse** of Word2Idx. It has the unique IDs as keys and their corresponding words(terms) as values

Padding:

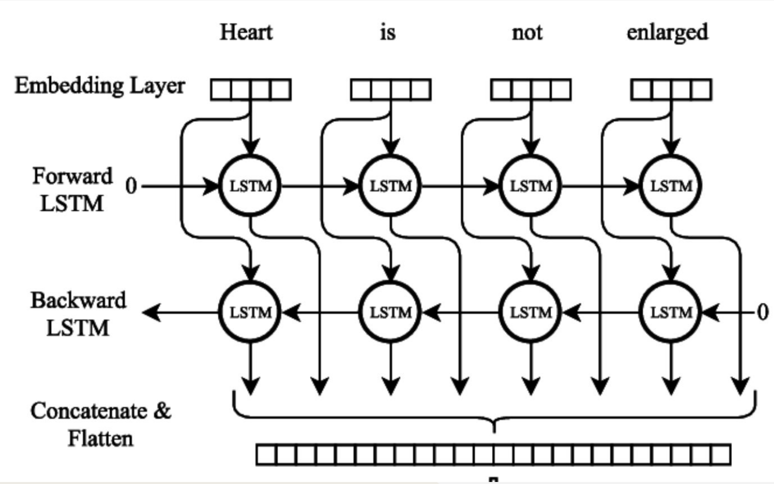
sequence_pad_sequences:It helps to ensure the **max length** and padding methods(**post padding**)

02

Model

Bi-lstm without pretrained weights

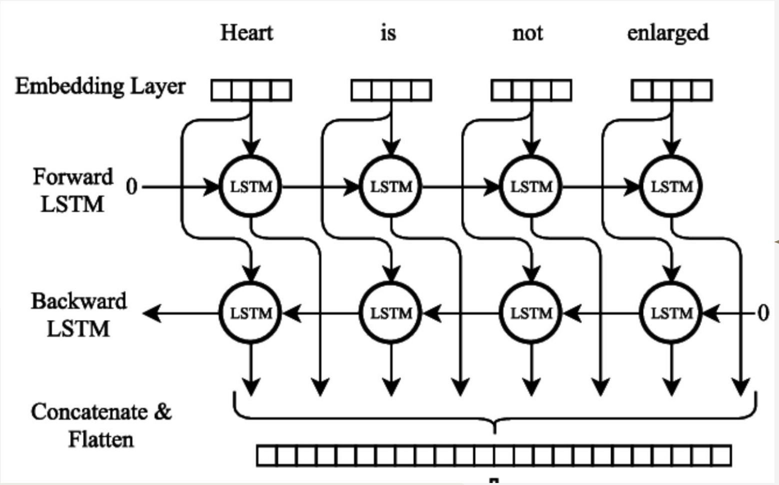
Bi-LSTM model structure



In this project, we used two models to classify named entities in text into pre-defined categories.

Firstly, we implemented a similar bidirectional LSTM model based on Zhiheng.H's sequence tagging paper.

Bi-LSTM model structure

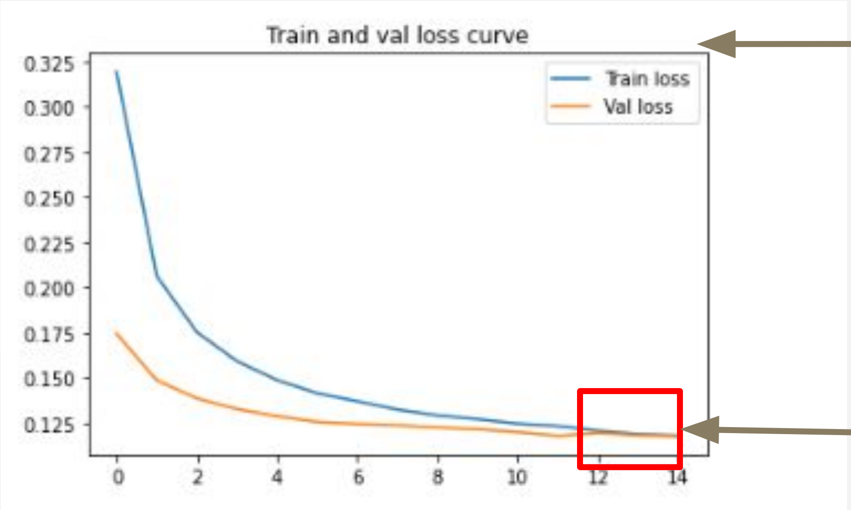


Embedding layer

Two bi-LSTM layer

Fully connected layer

Bi-LSTM model results



Train and validation loss curve

The stage to begin overfit

Bi-LSTM model results

	precision	recall	f1-score	support
B-art	0.00	0.00	0.00	0
B-eve	0.36	0.02	0.04	861
B-geo	0.89	0.81	0.85	8172
B-gpe	0.91	0.97	0.94	2963
B-nat	0.19	0.16	0.18	44
B-org	0.70	0.74	0.72	3766
B-per	0.80	0.80	0.80	3357
B-tim	0.87	0.90	0.88	3981
I-art	0.00	0.00	0.00	125
I-eve	0.20	0.18	0.19	44
I-geo	0.81	0.67	0.73	1778
I-gpe	0.47	0.79	0.59	19
I-nat	0.50	0.35	0.41	17
I-org	0.78	0.20	0.32	12852
I-per	0.84	0.03	0.05	106360
I-tim	0.76	0.78	0.77	1289
O	0.99	0.31	0.47	573772
PAD	0.00	0.00	0.00	0
accuracy			0.28	719400
macro avg	0.56	0.43	0.44	719400
weighted avg	0.96	0.28	0.41	719400

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

The testing accuracy we was **96.58 %.**

However, the model has a low F1 score.

Try pretrained model to improve the performance.

03

Model

Bert

BERT Basic

BERT (Bidirectional Encoder Representations from Transformers) is a by model proposed by researchers at Google AI Language. It presents state-of-the-art results in a wide variety of **NLP tasks**.

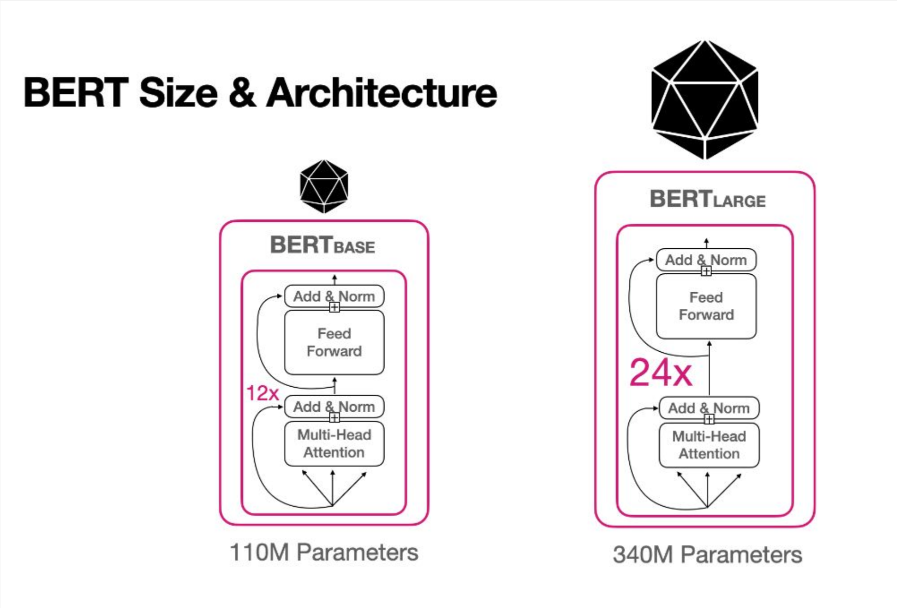
BERT’s key technical innovation is applying the **bidirectional training of Transformer**, a popular attention model, to language modelling. This is in contrast to previous efforts which looked at a text sequence either from left to right or combined left-to-right and right-to-left training.

BERT Keypoints

1. Large amounts of training data
   Wikipedia (~2.5B words) + Google Books Corpus (~800M words)
2. Masked Language Model
   Making word in sentence + Force bidirectionally
3. Next Sentence Prediction
   Sentence relationships
4. Transformers
   Attention + Massive parallelization

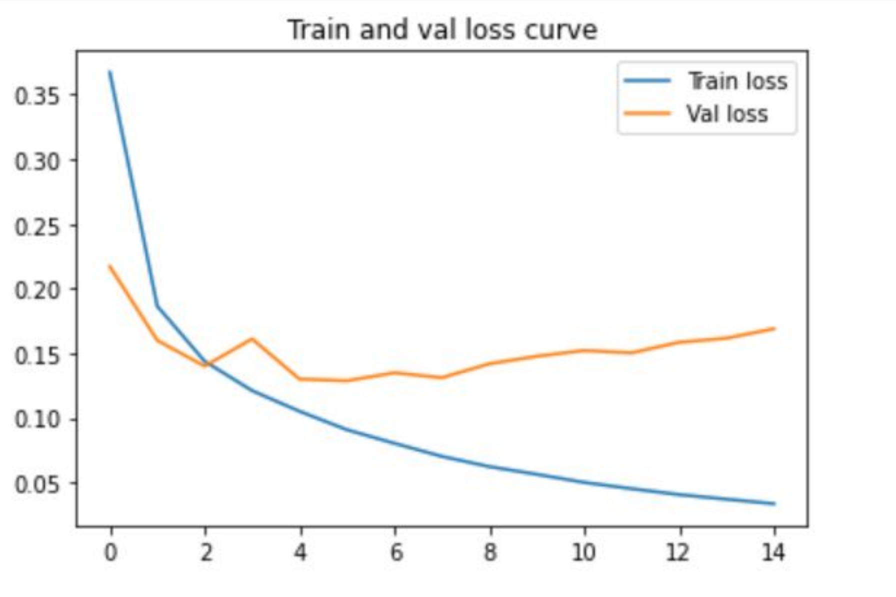
https://huggingface.co/blog/bert-101#1-what-is-bert-used-for

BERT model size & architecture



We used the BERT-base model for our task

Pre-trained BERT model results



Continue drop in training loss Vs
Fluctuate(Growth) trend in validation loss

→ A sign of overfitting

1. Model too complex
2. Pre-trained model
3. Too little dataset

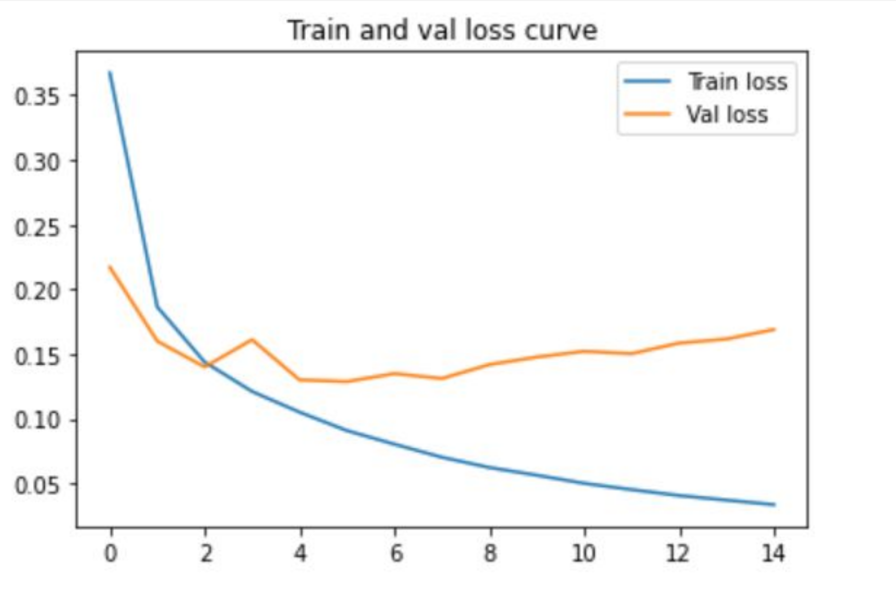
Improved BERT

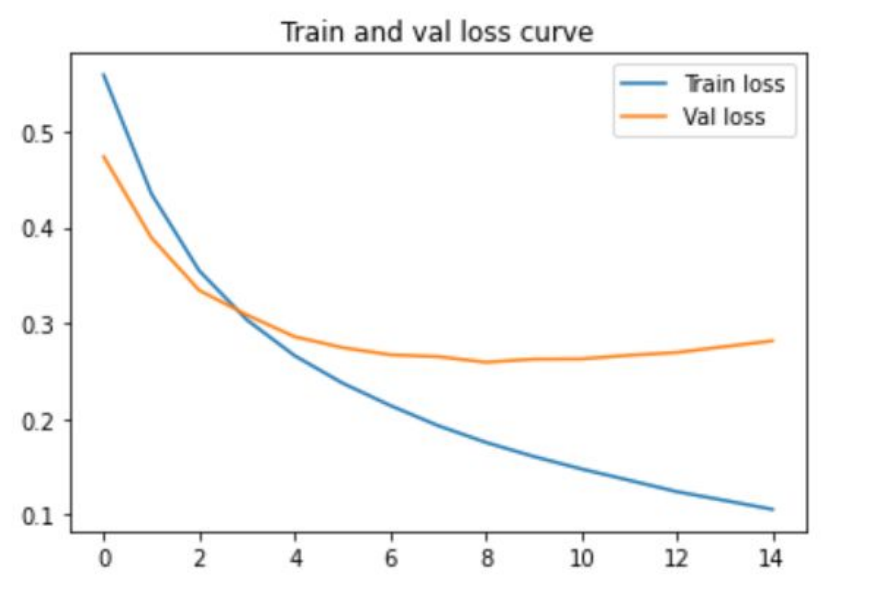
1. Without pre-trained
2. Only 3 layers

```
[ ]  model = BertForTokenClassification.from_pretrained("bert-base-uncased", num_labels=len(tag2idx))
     model.to(device)
```

```
[ ]  config = BertConfig(vocab_size_or_config_json_file= 30522, num_hidden_layers=3)
     model = BertForTokenClassification(config=config, num_labels=len(tag2idx))
     model.to(device)
```

Result Comparison





Test Loss: 0.169 I Test Acc: 92.40%

Test Loss: 0.276 I Test Acc: 95.11%

F1 Result Comparison

	precision	recall	f1-score	support
B-art	0.00	0.00	0.00	0
B-eve	0.36	0.02	0.04	861
B-geo	0.89	0.81	0.85	8172
B-gpe	0.91	0.97	0.94	2963
B-nat	0.19	0.16	0.18	44
B-org	0.70	0.74	0.72	3766
B-per	0.80	0.80	0.80	3357
B-tim	0.87	0.90	0.88	3981
I-art	0.00	0.00	0.00	125
I-eve	0.20	0.18	0.19	44
I-geo	0.81	0.67	0.73	1778
I-gpe	0.47	0.79	0.59	19
I-nat	0.50	0.35	0.41	17
I-org	0.78	0.20	0.32	12852
I-per	0.84	0.03	0.05	106360
I-tim	0.76	0.78	0.77	1289
O	0.99	0.31	0.47	573772
PAD	0.00	0.00	0.00	0
accuracy			0.28	719400
macro avg	0.56	0.43	0.44	719400
weighted avg	0.96	0.28	0.41	719400

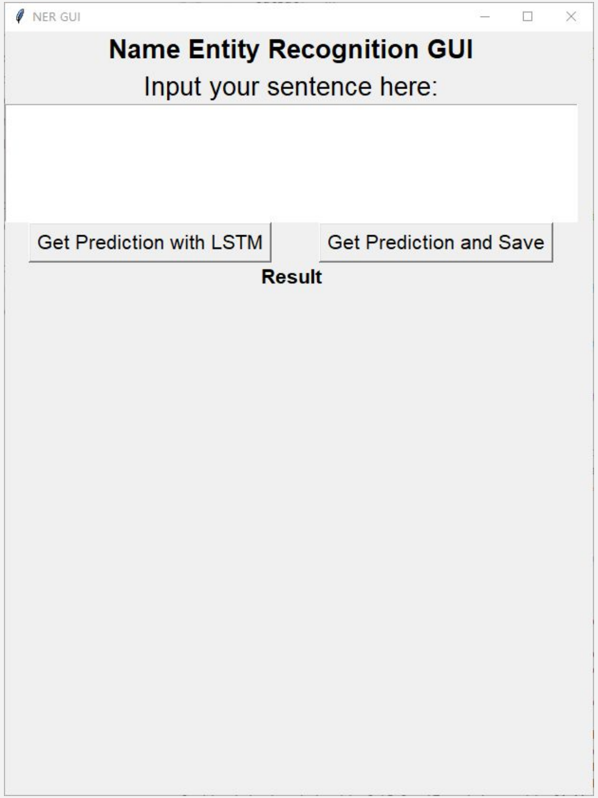
	precision	recall	f1-score	support
B-art	0.12	0.41	0.18	22
B-eve	0.17	0.34	0.23	32
B-geo	0.61	0.46	0.52	9968
B-gpe	0.77	0.28	0.41	8533
B-nat	0.24	0.48	0.32	23
B-org	0.55	0.30	0.39	7267
B-per	0.68	0.61	0.64	3752
B-tim	0.64	0.49	0.55	5305
I-art	0.00	0.00	0.00	12
I-eve	0.16	0.60	0.25	15
I-geo	0.52	0.44	0.48	1743
I-gpe	0.30	0.81	0.44	16
I-nat	0.12	0.33	0.18	3
I-org	0.51	0.43	0.47	3941
I-per	0.75	0.61	0.67	4230
I-tim	0.48	0.32	0.38	1909
O	0.97	0.99	0.98	672629
accuracy			0.95	719400
macro avg	0.45	0.46	0.42	719400
weighted avg	0.94	0.95	0.95	719400

04

GUI

Tkinter

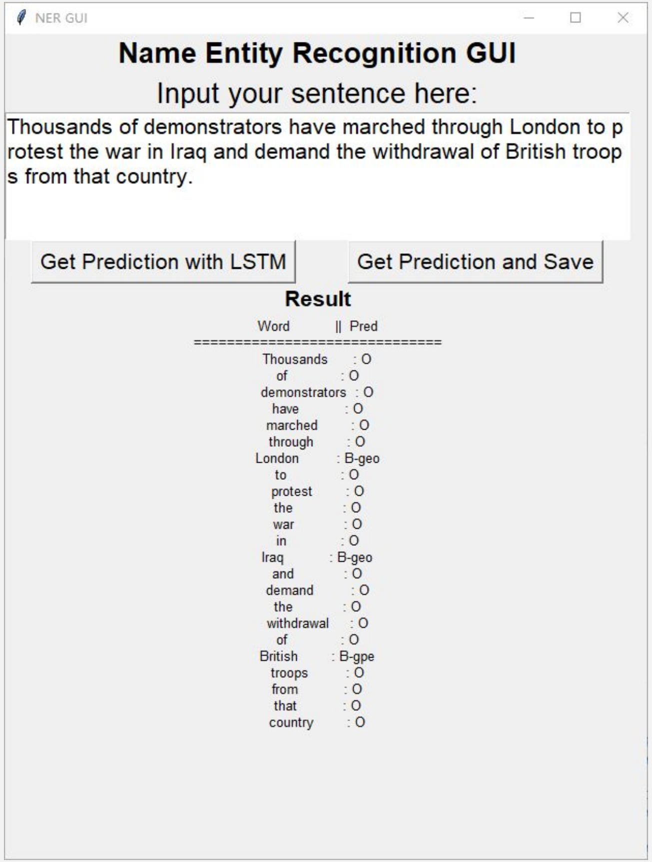
Design of GUI



The design choice:
Only have the function that get the output from pre-trained model with input sentence.

Elements:
The GUI is developed with the built- in package Tkinter. It has 4 main elements. Input box, 2 buttons and output space.
After inputting the sentence to the box, with the left button, the output is displayed with predictions from our LSTM model.
After inputting the sentence to the box, with the right button, the output is displayed with predictions from our LSTM model. Besides, a text file output.txt is generated within the same folder of GUI.py, the content inside is the output.

Sample output from the GUI



The left site is the sample output.
Note that only one sentence at each time. The space, “\n” and ”\r” at the end of the sentence will be automatically removed before passing to the model.

DEMO

THANKS