```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
np.random.seed(0)
plt.style.use("ggplot")
```

```
import tensorflow as tf
print('Tensorflow version:', tf.__version__)
print('GPU detected:', tf.config.list_physical_devices('GPU'))
```

```
Tensorflow version: 2.15.0
GPU detected: []
```

```
data = pd.read_csv("ner_dataset.csv", encoding="latin1")
data = data.fillna(method="ffill")
data.head(20)
```

DataFrame: data

View

DataFrame with shape (145102, 4)

	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
10	Sentence: 1	war	NN	O
11	Sentence: 1	in	IN	O
12	Sentence: 1	Iraq	NNP	B-geo
13	Sentence: 1	and	CC	O
14	Sentence: 1	demand	VB	O
15	Sentence: 1	the	DT	O
16	Sentence: 1	withdrawal	NN	O
17	Sentence: 1	of	IN	O
18	Sentence: 1	British	JJ	B-gpe
19	Sentence: 1	troops	NNS	O

```
print("Unique words in corpus:", data['Word'].nunique())
print("Unique tags in corpus:", data['Tag'].nunique())
```

```
Unique words in corpus: 13134
Unique tags in corpus: 17
```

```
words = list(set(data["Word"].values))
words.append("ENDPAD")
num_words = len(words)
```

```
tags = list(set(data["Tag"].values))
num_tags = len(tags)
```

```
class SentenceGetter(object):
    def __init__(self, data):
        self.n_sent = 1
        self.data = data
        self.empty = False
        agg_func = lambda s: [(w, p, t) for w, p, t in zip(s["Word"].values.tolist(),
                                                           s["POS"].values.tolist(),
                                                           s["Tag"].values.tolist())]
        self.grouped = self.data.groupby("Sentence #").apply(agg_func)
        self.sentences = [s for s in self.grouped]

    def get_next(self):
        try:
            s = self.grouped["Sentence: {}".format(self.n_sent)]
            self.n_sent += 1
            return s
        except:
            return None
```

DataFrame: data

View

DataFrame with shape (145102, 4)

```
getter = SentenceGetter(data)
sentences = getter.sentences


sentences[0]
```

```
[('Thousands', 'NNS', 'O'),
 ('of', 'IN', 'O'),
 ('demonstrators', 'NNS', 'O'),
 ('have', 'VBP', 'O'),
 ('marched', 'VBN', 'O'),
 ('through', 'IN', 'O'),
 ('London', 'NNP', 'B-geo'),
 ('to', 'TO', 'O'),
 ('protest', 'VB', 'O'),
 ('the', 'DT', 'O'),
 ('war', 'NN', 'O'),
 ('in', 'IN', 'O'),
 ('Iraq', 'NNP', 'B-geo'),
 ('and', 'CC', 'O'),
 ('demand', 'VB', 'O'),
 ('the', 'DT', 'O'),
 ('withdrawal', 'NN', 'O'),
 ('of', 'IN', 'O'),
 ('British', 'JJ', 'B-gpe'),
 ('troops', 'NNS', 'O'),
 ('from', 'IN', 'O'),
 ('that', 'DT', 'O'),
 ('country', 'NN', 'O'),
 ('.', '.', 'O')]
```

```
word2idx = {w: i + 1 for i, w in enumerate(words)}
tag2idx = {t: i for i, t in enumerate(tags)}


word2idx
```

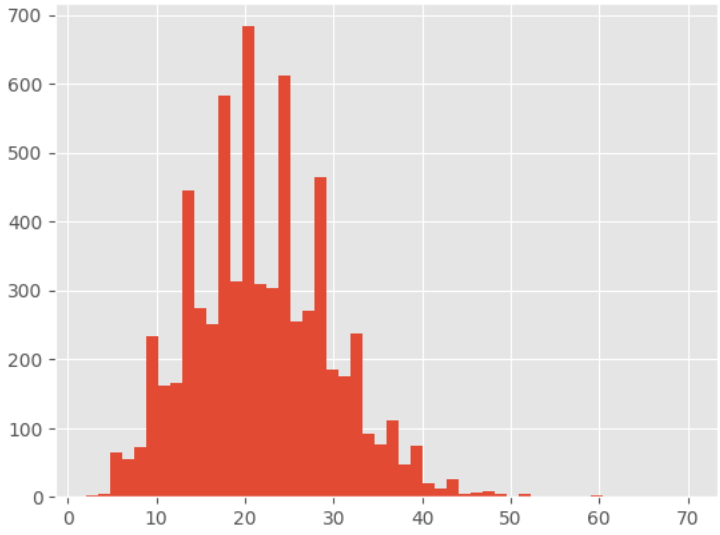
```
'significantly': 786,
'statistics': 787,
'Pemex': 788,
'surpassing': 789,
'sperm': 790,
'apostolic': 791,
'Aysegul': 792,
'Ante': 793,
'Tanzania': 794,
'Hague': 795,
'behalf': 796,
'exile': 797,
'Jong': 798,
'Rafael': 799,
'Spokeswoman': 800,
'suspicions': 801,
'fuel-efficient': 802,
'Philippine': 803,
'rear': 804,
'Earnings': 805,
'try': 806,
'biographies': 807,
'Arcega': 808,
'repeated': 809,
'unacceptable': 810,
'attempting': 811,
'Musicians': 812,
```

DataFrame: data

View

DataFrame with shape (145102, 4)

```
plt.hist([len(s) for s in sentences], bins=50)
plt.show()
```




```
from tensorflow.keras.preprocessing.sequence import pad_sequences

max_len = 50

X = [[word2idx[w[0]] for w in s] for s in sentences]
X = pad_sequences(maxlen=max_len, sequences=X, padding="post", value=num_words-1)

y = [[tag2idx[w[2]] for w in s] for s in sentences]
y = pad_sequences(maxlen=max_len, sequences=y, padding="post", value=tag2idx["O"])


from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)


from tensorflow.keras import Model, Input
from tensorflow.keras.layers import LSTM, Embedding, Dense
from tensorflow.keras.layers import TimeDistributed, SpatialDropout1D, Bidirectional


input_word = Input(shape=(max_len,))
model = Embedding(input_dim=num_words, output_dim=50, input_length=max_len)(input_word)
model = SpatialDropout1D(0.1)(model)
model = Bidirectional(LSTM(units=100, return_sequences=True, recurrent_dropout=0.1))(model)
out = TimeDistributed(Dense(num_tags, activation="softmax"))(model)
model = Model(input_word, out)
model.summary()
```

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
```

```
 =================================================================
 input_1 (InputLayer)        [(None, 50)]              0

 embedding (Embedding)       (None, 50, 50)            656750

 spatial_dropout1d (Spatial  (None, 50, 50)            0
 Dropout1D)

 bidirectional (Bidirection  (None, 50, 200)           120800
 al)

 time_distributed (TimeDist  (None, 50, 17)            3417
 ributed)

=================================================================
Total params: 780967 (2.98 MB)
Trainable params: 780967 (2.98 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

DataFrame: data

View

DataFrame with shape (145102, 4)

```
model.compile(optimizer="adam",
              loss="sparse_categorical_crossentropy",
              metrics=["accuracy"])
```

```
!pip install livelossplot
```

```
Collecting livelossplot
  Downloading livelossplot-0.5.5-py3-none-any.whl (22 kB)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from livelossplot) (3.7.1)
Requirement already satisfied: bokeh in /usr/local/lib/python3.10/dist-packages (from livelossplot) (3.3.4)
Requirement already satisfied: Jinja2>=2.9 in /usr/local/lib/python3.10/dist-packages (from bokeh->livelossplot) (3.1.3)
Requirement already satisfied: contourpy>=1 in /usr/local/lib/python3.10/dist-packages (from bokeh->livelossplot) (1.2.1)
Requirement already satisfied: numpy>=1.16 in /usr/local/lib/python3.10/dist-packages (from bokeh->livelossplot) (1.25.2)
Requirement already satisfied: packaging>=16.8 in /usr/local/lib/python3.10/dist-packages (from bokeh->livelossplot) (24.0)
Requirement already satisfied: pandas>=1.2 in /usr/local/lib/python3.10/dist-packages (from bokeh->livelossplot) (2.0.3)
Requirement already satisfied: pillow>=7.1.0 in /usr/local/lib/python3.10/dist-packages (from bokeh->livelossplot) (9.4.0)
Requirement already satisfied: PyYAML>=3.10 in /usr/local/lib/python3.10/dist-packages (from bokeh->livelossplot) (6.0.1)
Requirement already satisfied: tornado>=5.1 in /usr/local/lib/python3.10/dist-packages (from bokeh->livelossplot) (6.3.3)
Requirement already satisfied: xyzservices>=2021.09.1 in /usr/local/lib/python3.10/dist-packages (from bokeh->livelossplot) (2024.4
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->livelossplot) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->livelossplot) (4.51.0
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->livelossplot) (1.4.5)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->livelossplot) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib->livelossplot) (2.8
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from Jinja2>=2.9->bokeh->livelossplot) (
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.2->bokeh->livelossplot) (202
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.2->bokeh->livelossplot) (2
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib->liveloss
Installing collected packages: livelossplot
Successfully installed livelossplot-0.5.5
```

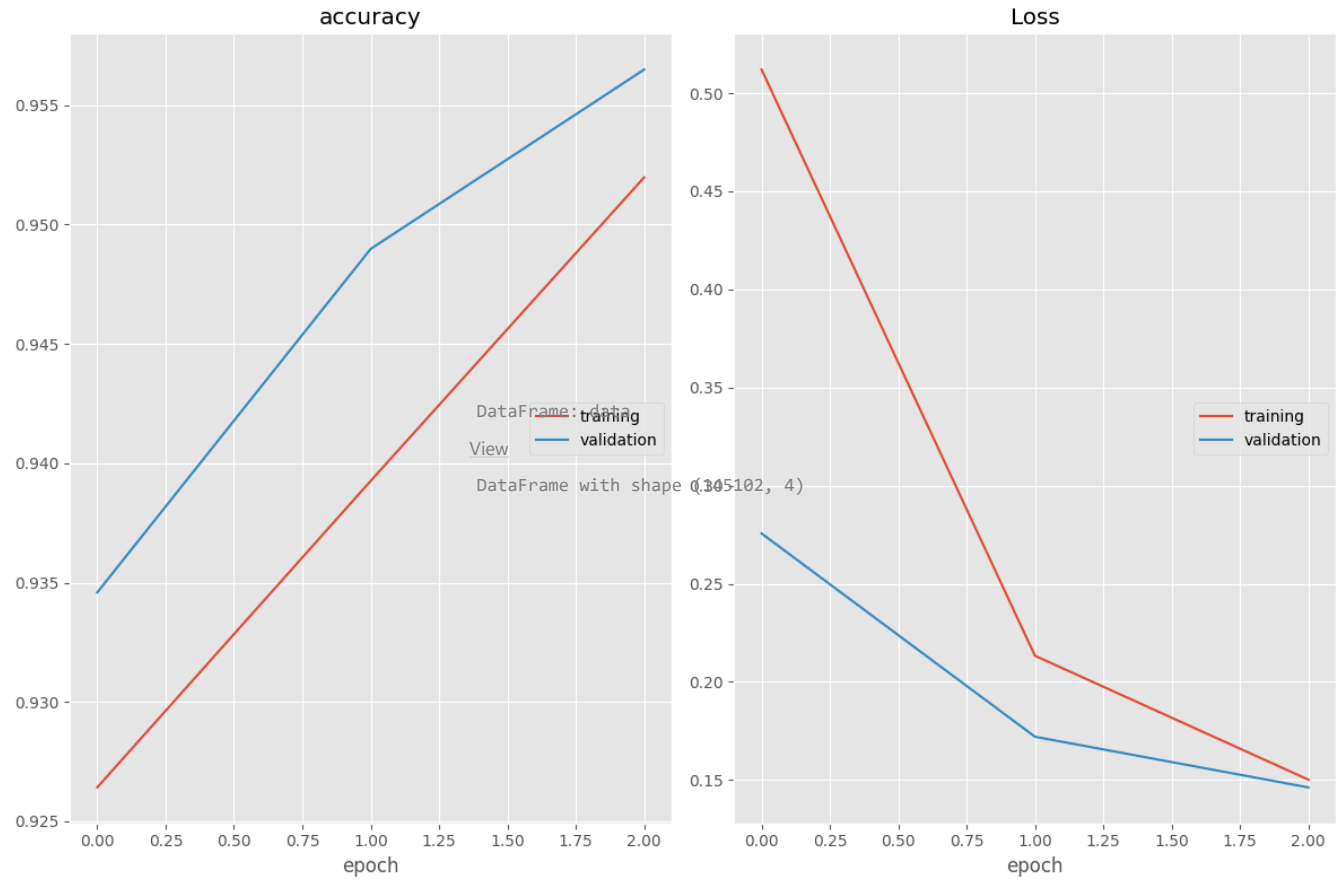
```
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping
from livelossplot.tf_keras import PlotLossesCallback


chkpt = ModelCheckpoint("model_weights.h5", monitor='val_loss',verbose=1, save_best_only=True, save_weights_only=True, mode='min')

early_stopping = EarlyStopping(monitor='val_accuracy', min_delta=0, patience=1, verbose=0, mode='max', baseline=None, restore_best_weig

callbacks = [PlotLossesCallback(), chkpt, early_stopping]

history = model.fit(
    x=x_train,
    y=y_train,
    validation_data=(x_test,y_test),
    batch_size=32,
    epochs=3,
    callbacks=callbacks,
    verbose=1
)
```




```
accuracy
        training                (min:    0.926, max:    0.952, cur:    0.952)
        validation              (min:    0.935, max:    0.956, cur:    0.956)
Loss
        training                (min:    0.150, max:    0.512, cur:    0.150)
        validation              (min:    0.146, max:    0.276, cur:    0.146)

Epoch 3: val_loss improved from 0.17200 to 0.14617, saving model to model_weights.h5
166/166 [==============================] - 39s 237ms/step - loss: 0.1501 - accuracy: 0.9520 - val_loss: 0.1462 - val_accuracy: 0.95
```

```
i = np.random.randint(0, x_test.shape[0])
p = model.predict(np.array([x_test[i]]))
p = np.argmax(p, axis=-1)
y_true = y_test[i]
print("{:15}{:5}\t {}\n".format("Word", "True", "Pred"))
print("-" *30)
for w, true, pred in zip(x_test[i], y_true, p[0]):
    print("{:15}{}\t{}".format(words[w-1], tags[true], tags[pred]))
```

```
1/1 [==============================] - 1s 1s/step
Word           True      Pred

------------------------------
Togo           B-geo     I-per
's             O         O
election       O         O
commission     O         O
says           O         O
ruling         O         O
party          O         O
candidate      O         O
Faure          B-per     O
Gnassingbe     I-per     I-org
is             O         O
the            O         O
winner         O         O
of             O         O
Sunday         B-tim     B-geo
's             O         O
presidential   O         O
election       O         O
.              O         O
```

```
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
Bystanders     0        0
```

DataFrame: data

View

DataFrame with shape (145102, 4)