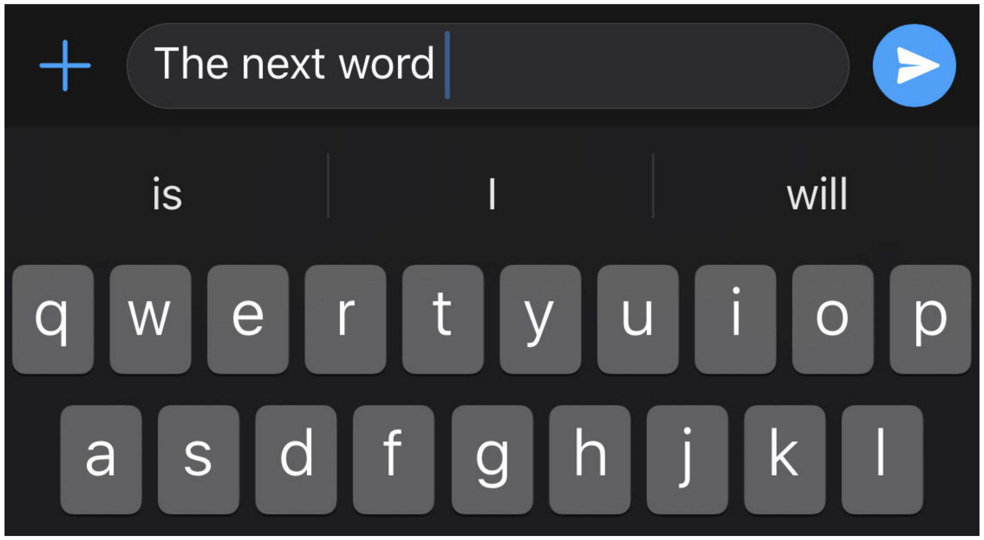
# Lab Assignment

## Assignment 1

**Using LSTM  try to train next word prediction for dataset data.txt.**

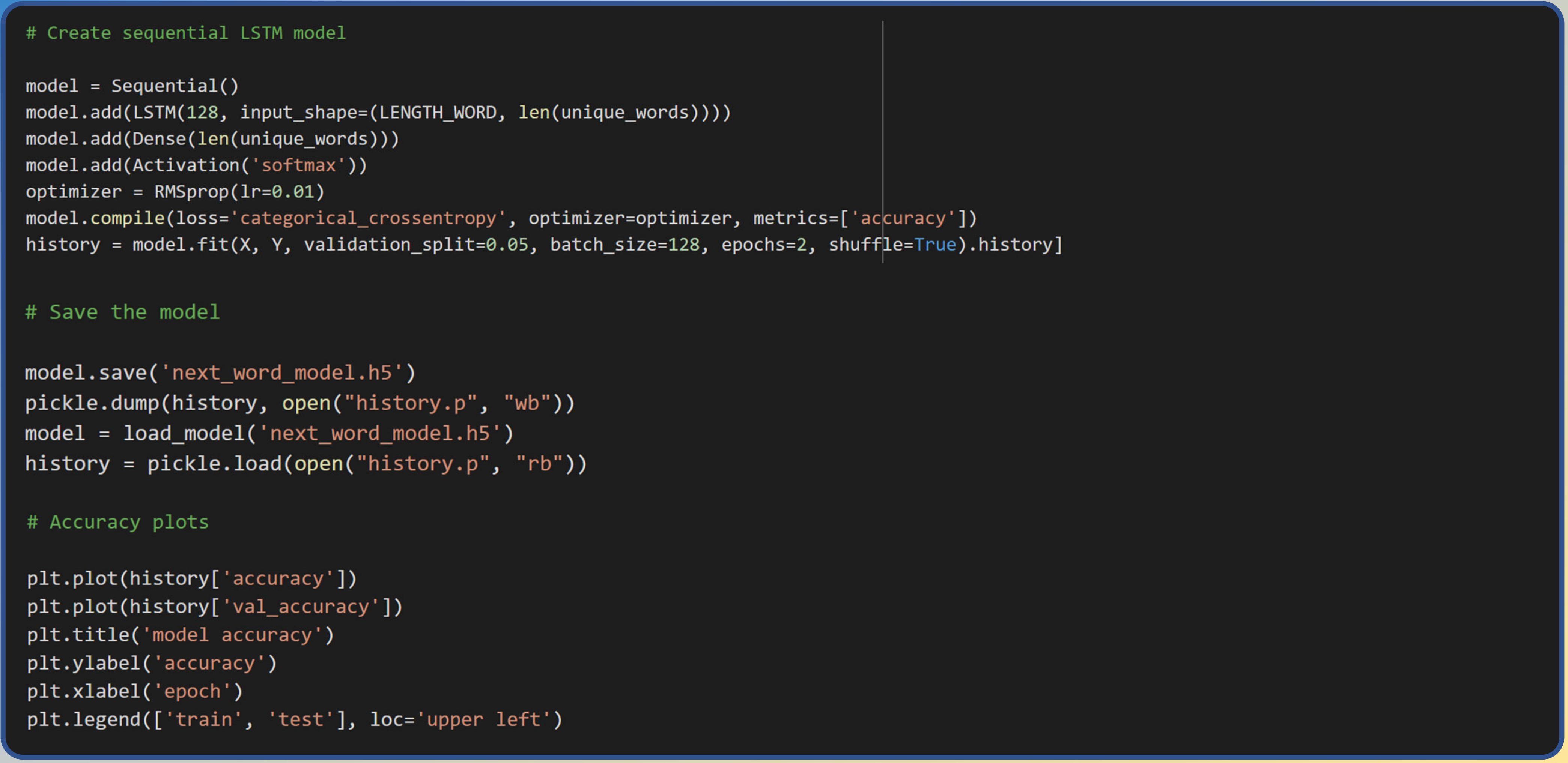


# Solution

```python
# Import required modules

import numpy as np
import heapq
import matplotlib.pyplot as plt
from nltk.tokenize import RegexpTokenizer
from keras.models import Sequential, load_model
from keras.layers.core import Dense, Activation
from keras.layers import LSTM
import pickle
from keras.optimizers import RMSprop


# Import dataset

path = 'data.txt'
text = open(path).read().lower()
print('Length of the corpus is: :', len(text))

print(text[:100])

# Tokenize

tokenizer = RegexpTokenizer(r'\w+')
words = tokenizer.tokenize(text)
```

```python
# Get unique words and index dictionary

unique_words = np.unique(words)
unique_word_index = dict((c, i) for i, c in enumerate(unique_words))

# Create a set of next and previous words

LENGTH_WORD = 5
next_words = []
prev_words = []
for j in range(len(words) - LENGTH_WORD):
     prev_words.append(words[j:j + LENGTH_WORD])
     next_words.append(words[j + LENGTH_WORD])
print(prev_words[0])
print(next_words[0])


# Create X and Y to store word projections

X = np.zeros((len(prev_words), LENGTH_WORD, len(unique_words)), dtype=bool)
Y = np.zeros((len(next_words), len(unique_words)), dtype=bool)
for i, each_words in enumerate(prev_words):
   for j, each_word in enumerate(each_words):
        X[i, j, unique_word_index[each_word]] = 1
   Y[i, unique_word_index[next_words[i]]] = 1
```

## Solution



```
# Create sequential LSTM model

model = Sequential()
model.add(LSTM(128, input_shape=(LENGTH_WORD, len(unique_words))))
model.add(Dense(len(unique_words)))
model.add(Activation('softmax'))
optimizer = RMSprop(lr=0.01)
model.compile(loss='categorical_crossentropy', optimizer=optimizer, metrics=['accuracy'])
history = model.fit(X, Y, validation_split=0.05, batch_size=128, epochs=2, shuffle=True).history]


# Save the model

model.save('next_word_model.h5')
pickle.dump(history, open("history.p", "wb"))
model = load_model('next_word_model.h5')
history = pickle.load(open("history.p", "rb"))

# Accuracy plots

plt.plot(history['accuracy'])
plt.plot(history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
```

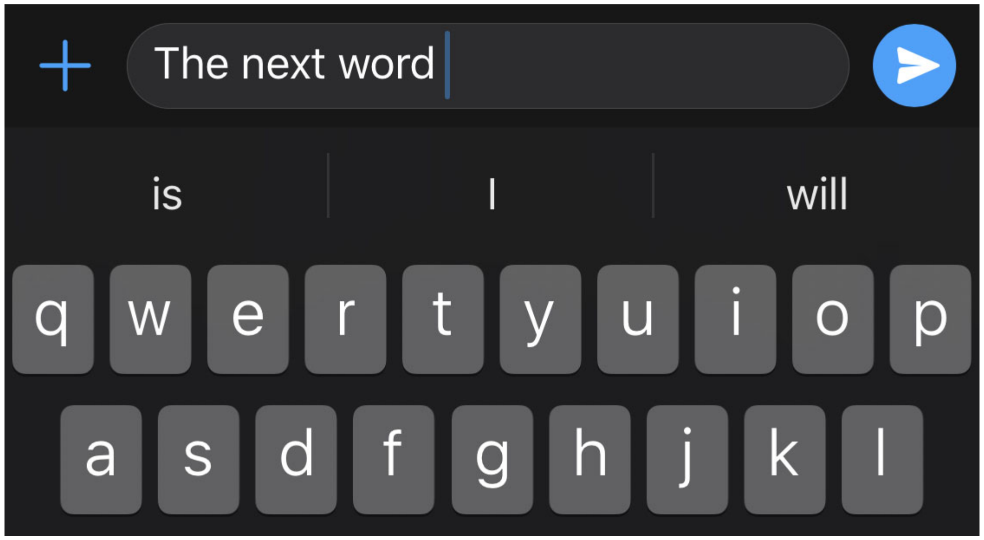
## Solution

```python
# Loss plots

plt.plot(history['loss'])
plt.plot(history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
```

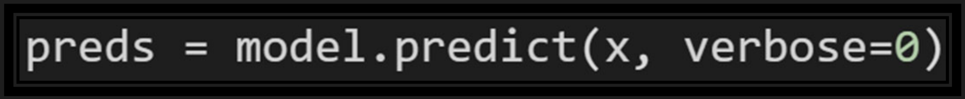
# Lab Assignment

## Assignment 2

Using LSTM try to test next word prediction for dataset data.txt.



## Solution

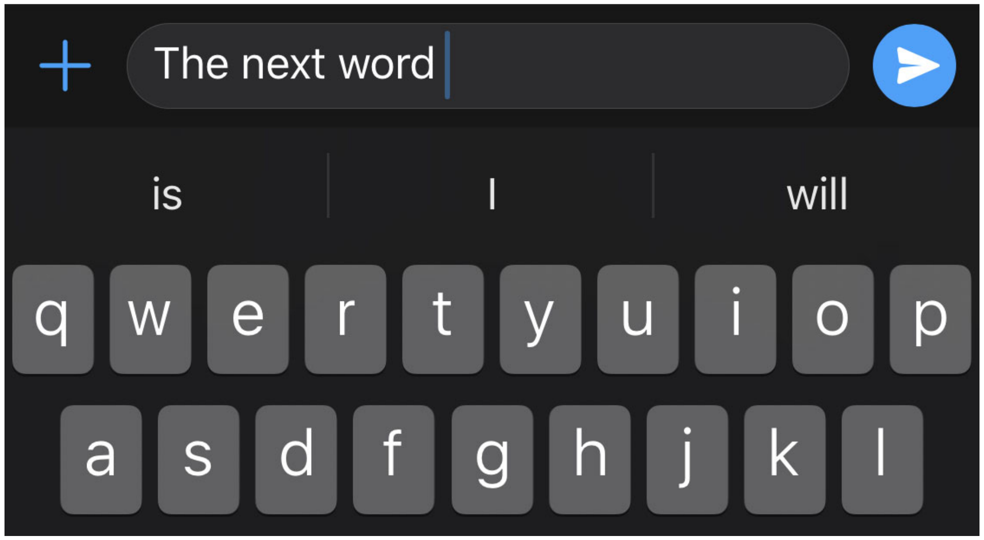


```
preds = model.predict(x, verbose=0)
```

# Lab Assignment

## Assignment 3

Using **LSTM** try to build **next word prediction** for dataset **data.txt** using a **pretrained embedding**.


The next word
is
I
will
q w e r t y u i o p
a s d f g h j k l