

LSTM 모델을 이용한 가짜 뉴스 판별기

21700165 김주환
21900633 장해원

목차

프로젝트
선정 계기

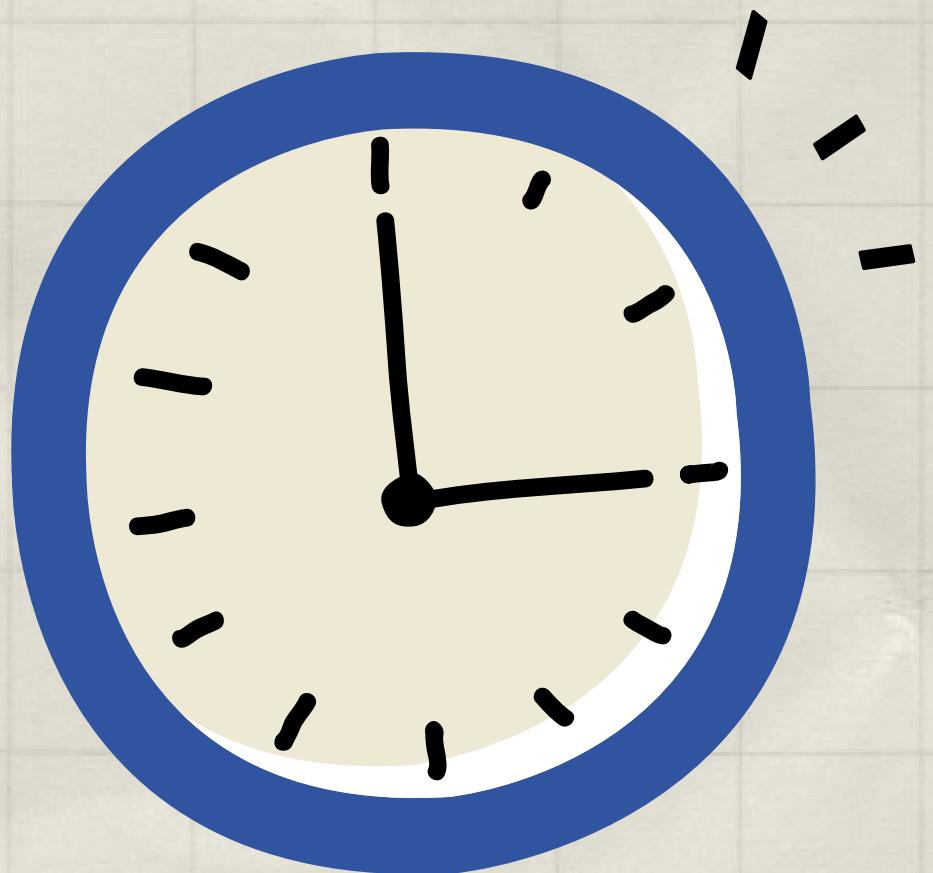
사용모델

학습과정

학습 결과

프로젝트
소감
&
기대 효과

데모



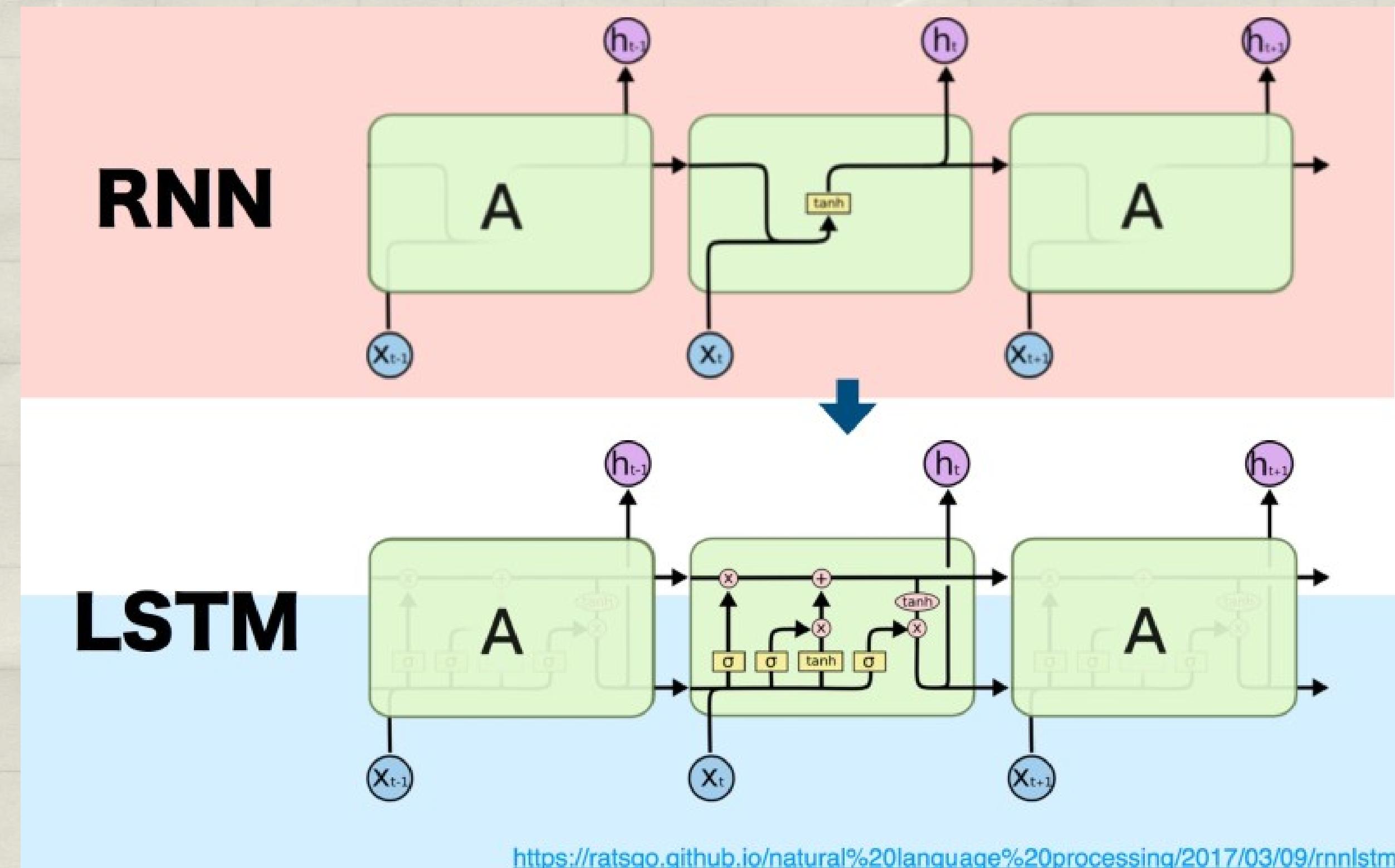
프로젝트 선정 계기

관심이 곧 돈이 되는 사회

클릭을 위한 가짜 뉴스가 폭발적으로 증가

가짜 뉴스로 의한 피해 방지와 예방

사용 모델



<https://ratsgo.github.io/natural%20language%20processing/2017/03/09/rnnlstm/>

학습과정

뉴스 테이터 분류

```
xtr = np.load('./xtr_shuffled.npy', allow_pickle=True)
xte = np.load('./xte_shuffled.npy', allow_pickle=True)
y_train = np.load('./ytr_shuffled.npy', allow_pickle=True)
y_test = np.load('./yte_shuffled.npy', allow_pickle=True)
```

학습과정

데이터 정제

```
text = re.sub(r"[^A-Za-z0-9^,!.\\/'+=]", " ", text)
text = text.lower().split()
stops = set(stopwords.words("english"))
text = [w for w in text if not w in stops]
text = " ".join(text)
```

학습과정

워드 임베딩

```
embedding_vecor_length = 32
model = Sequential()
model.add(Embedding(top_words+2, embedding_vecor_length, input_length=max_review_length))
model.add(LSTM(100))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
print(model.summary())
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=epoch_num, batch_size=batch_size)
```

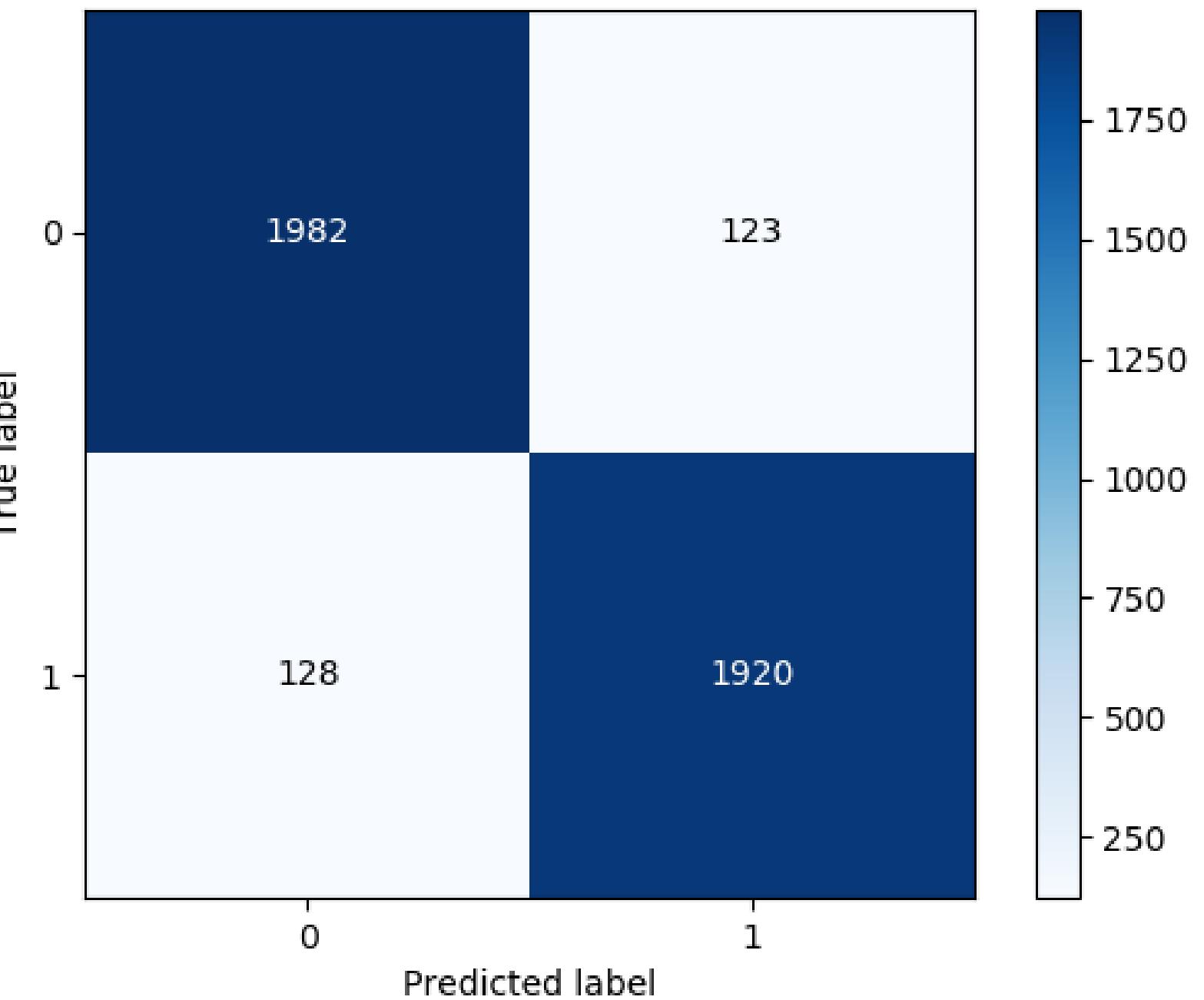
학습과정

워드 임베딩

```
embedding_vecor_length = 32
model = Sequential()
model.add(Embedding(top_words+2, embedding_vecor_length, input_length=max_review_length))
model.add(LSTM(100))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
print(model.summary())
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=epoch_num, batch_size=batch_size)
```

학습 결과

Confusion Matrix



```
None
Epoch 1/5
251/251 [=====] - 184s 726ms/step - loss: 0.3324 - accuracy: 0.8611 - val_loss: 0.1705 - val_accuracy: 0.9415
Epoch 2/5
251/251 [=====] - 180s 719ms/step - loss: 0.1231 - accuracy: 0.9585 - val_loss: 0.1458 - val_accuracy: 0.9487
Epoch 3/5
251/251 [=====] - 190s 756ms/step - loss: 0.0887 - accuracy: 0.9719 - val_loss: 0.1856 - val_accuracy: 0.9278
Epoch 4/5
251/251 [=====] - 170s 679ms/step - loss: 0.1235 - accuracy: 0.9510 - val_loss: 0.1608 - val_accuracy: 0.9502
Epoch 5/5
251/251 [=====] - 179s 715ms/step - loss: 0.1332 - accuracy: 0.9552 - val_loss: 0.2008 - val_accuracy: 0.9201
Accuracy= 92.01%
```

프로젝트 소감

어려웠던 점

자연어 처리 학습 과정

단어를 뺏기화 시킨다는 점

LSTM 모델의 이해

기대 효과

거름망의 역할

가짜 뉴스의 확산 방지

가짜 뉴스로 파생되는 피해방지

데모 영상

```
# Create the model
embedding_vecor_length = 32
model = Sequential()
model.add(Embedding(top_words+2, embedding_vecor_length, input_length=max_review_length))
model.add(LSTM(100))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
print(model.summary())
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=epoch_num, batch_size=batch_size)

# Final evaluation of the model
scores = model.evaluate(X_test, y_test, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))

Model: "sequential"
Layer (type)          Output Shape         Param #
=====
embedding (Embedding) (None, 500, 32)      160064
lstm (LSTM)           (None, 100)          53200
dense (Dense)         (None, 1)           101
=====
Total params: 213,365
Trainable params: 213,365
Non-trainable params: 0

None
Epoch 1/5
251/251 [=====] - 115s 451ms/step - loss: 0.3108 - accuracy: 0.8654 - val_loss: 0.3533 - val_accuracy: 0.8555
Epoch 2/5
251/251 [=====] - 110s 439ms/step - loss: 0.1504 - accuracy: 0.9479 - val_loss: 0.1570 - val_accuracy: 0.9441
Epoch 3/5
251/251 [=====] - 110s 440ms/step - loss: 0.1080 - accuracy: 0.9659 - val_loss: 0.1638 - val_accuracy: 0.9453
Epoch 4/5
209/251 [=====],....] - ETA: 17s - loss: 0.0907 - accuracy: 0.9698

# Draw the confusion matrix
y_pred = model.predict_classes(X_test)
plot_cmat(y_test, y_pred)
```