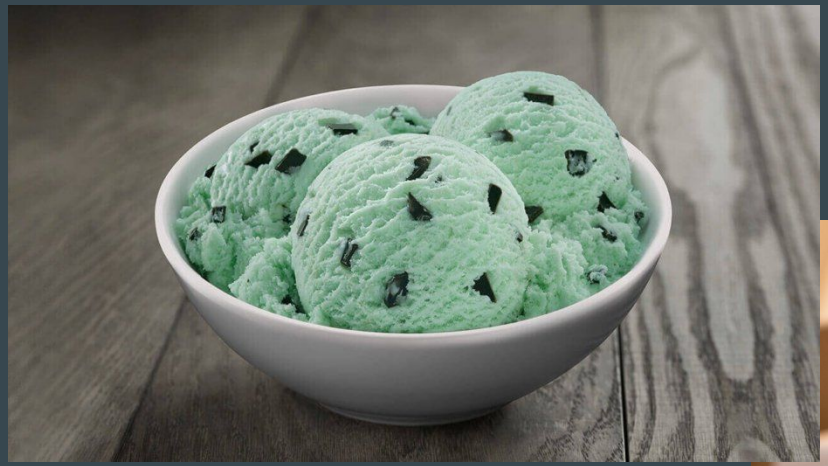
감정분석을 이용한 여론조사

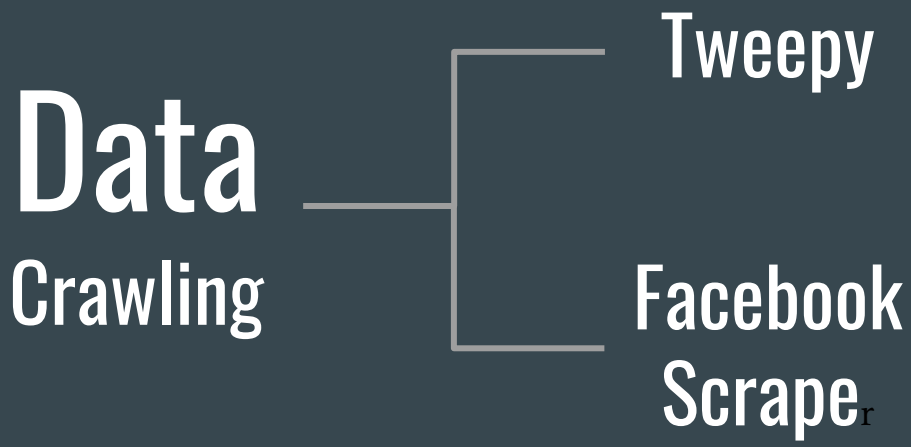
21700083 김도윤
22000168 김정욱



호불호

감정분석을 이용한 여론조사

예비보고서에서 바뀐점



예비보고서에서 바뀐점

3k -> 1.6 million

인공지능의 분야

자연어 처리

Natural Language Processing

Sentiment Analysis

코드 설명

model.ipynb

comment_scraper.py

app.py

/input

    training.1600000.processed.noemoticon.csv

/webdrivers

    chromedriver

/templates

    index.html

/static/css

    style.css

https://github.com/dodoyoon/SentimentAnalysis

kaggle

https://www.kaggle.com/kazanova/sentiment140

model.ipynb

Dataset

‘Sentiment140 dataset with 1.6 million tweets’ from kaggle

```
dataset_filename = os.listdir("./input")[0]
print(dataset_filename)
dataset_path = os.path.join(".","input",dataset_filename)
print("Open file:", dataset_path)
df = pd.read_csv(dataset_path, encoding =DATASET_ENCODING , names=DATASET_COLUMNS)
```

```
training.1600000.processed.noemoticon.csv
Open file: ./input/training.1600000.processed.noemoticon.csv
```

```
print("Dataset size:", len(df))
```

```
Dataset size: 1600000
```

```
df.head(5)
```

	target	ids	date	flag	user	text
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
2	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
3	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
4	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....

Pre-process dataset (데이터 전처리)

Remove link, user, special characters: stopwords 제거, stemming

stopwords(불용어): 코퍼스(말뭉치)에 자주 나타나지만 학습에 기여하지 않는 단어

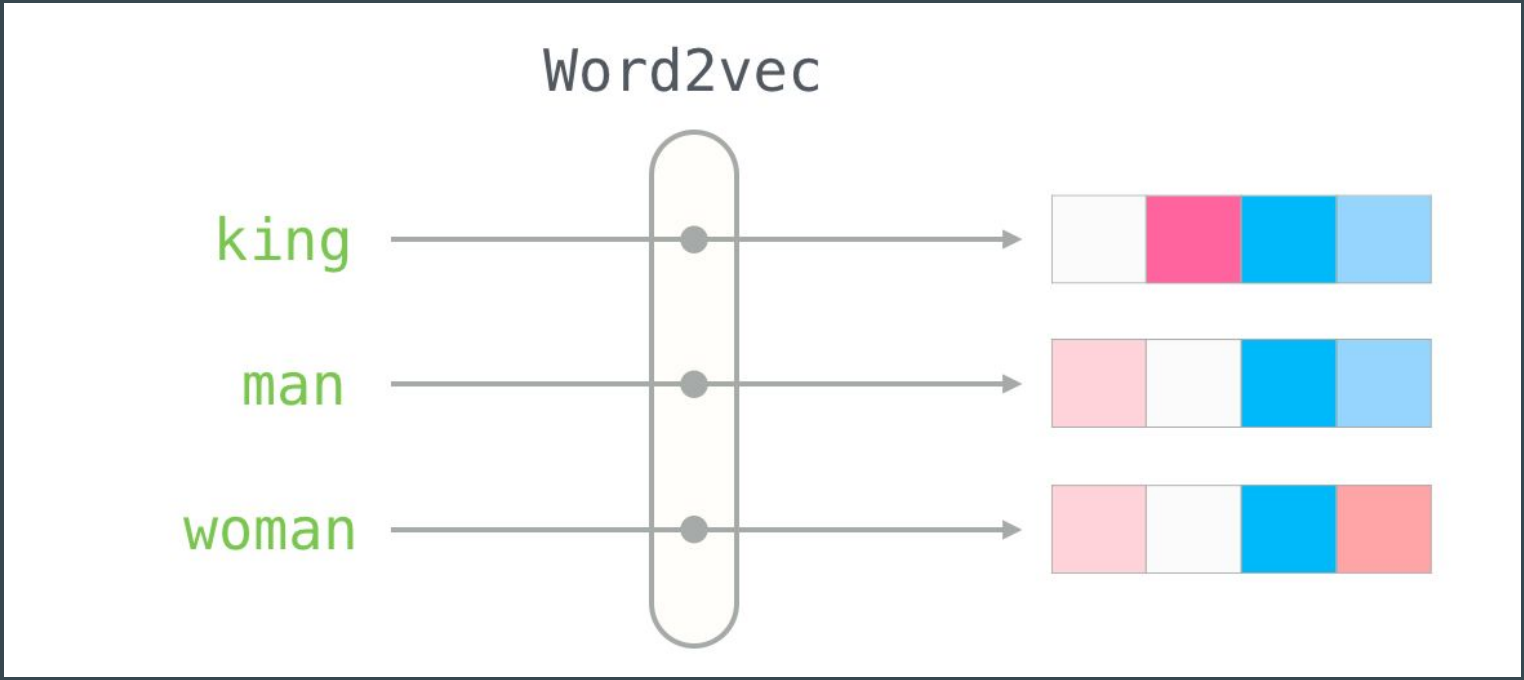
e.g. 조사, 접미사, i, me, my, it, this, that, is, are

stemming: 어형이 변형된 단어로부터 그 단어의 어간 분리

e.g. running, runs, run → run

예제 및 설명 출처:
https://programmers.co.kr/learn/courses/21/lessons/1694

Word Embedding: word2vec

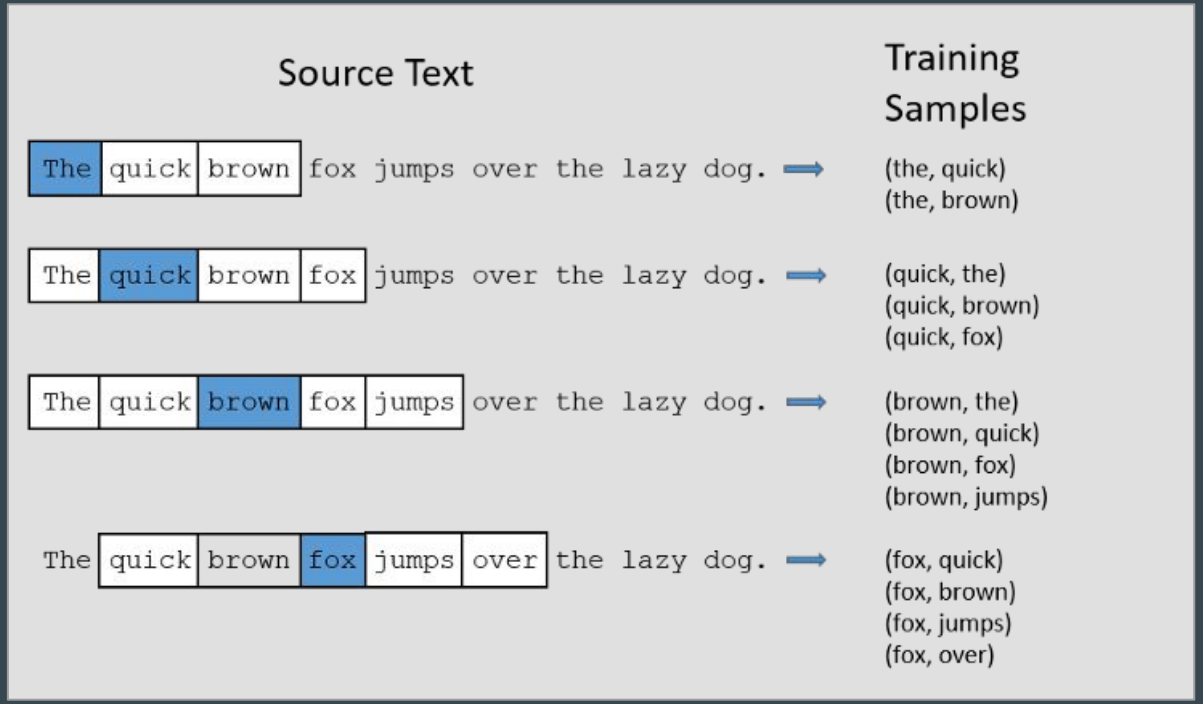


사진출처 : http://jalammar.github.io/illustrated-word2vec/

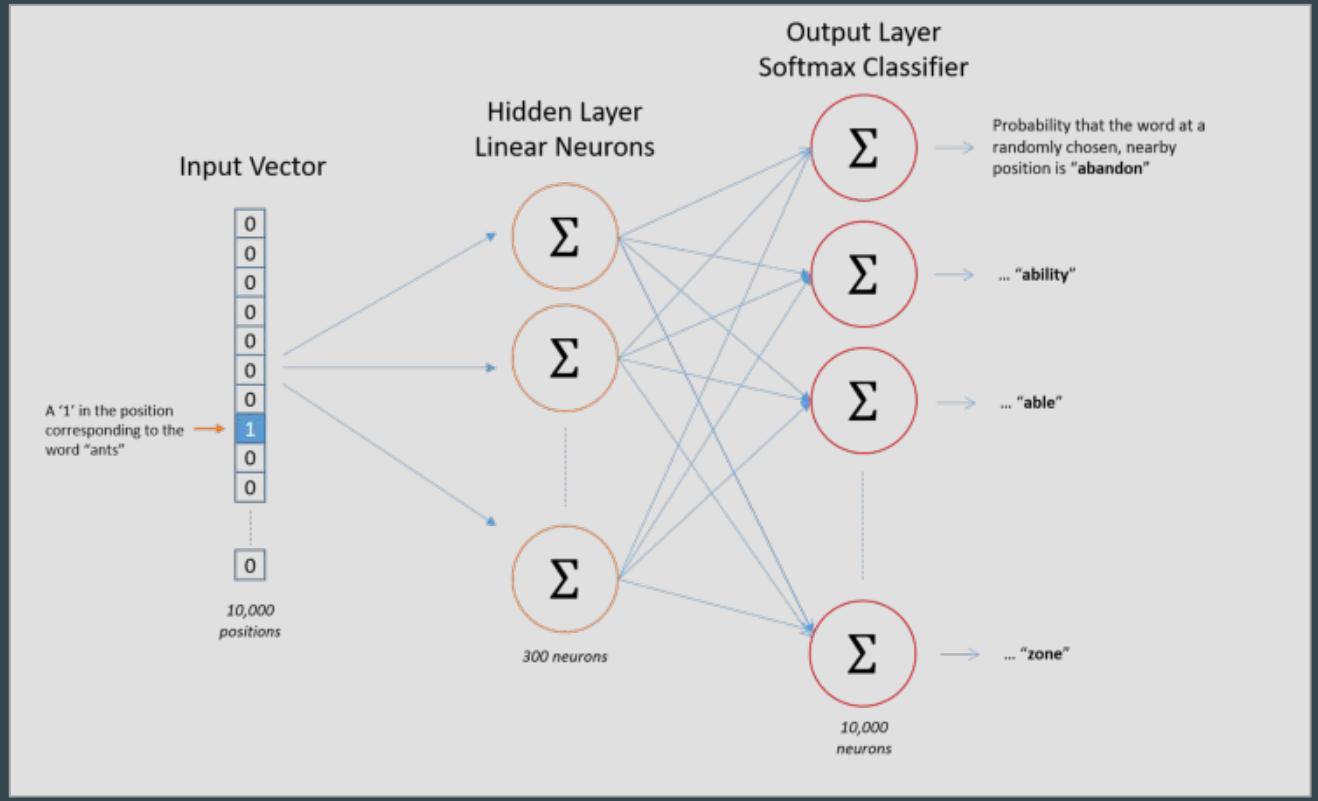
설명 출처: https://medium.com/datadriveninvestor/word2vec-skip-gram-model-explained-383fa6ddc4ae

word2vec: skip-gram

"The quick brown fox jumps over the lazy dog"



word2vec: skip-gram

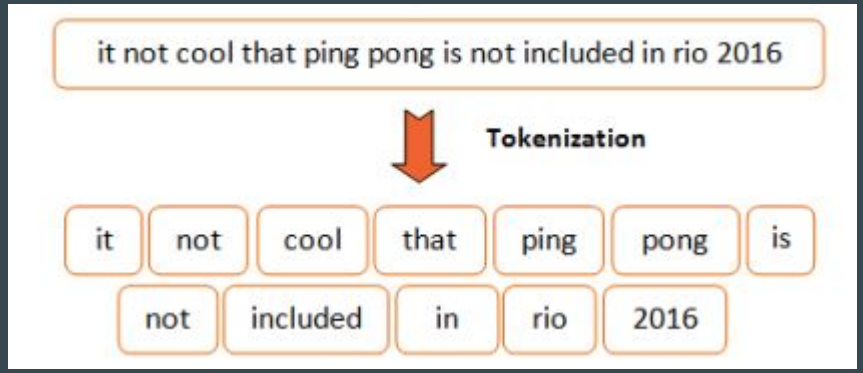


```
1 w2v_model.most_similar("love")
```

```
[('luv', 0.5688667297363281),
 ('loves', 0.5516188144683838),
 ('loved', 0.550737738609314),
 ('adore', 0.5176335573196411),
 ('amazing', 0.5148882269859314),
 ('looove', 0.4862058162689209),
 ('awesome', 0.4775022268295288),
 ('loveee', 0.46532952785491943),
 ('lovee', 0.4433733820915222),
 ('loooove', 0.4388321042060852)]
```

사진 출처: https://www.kaggle.com/arunrk7/nlp-beginner-text-classification-using-lstm

Tokenization

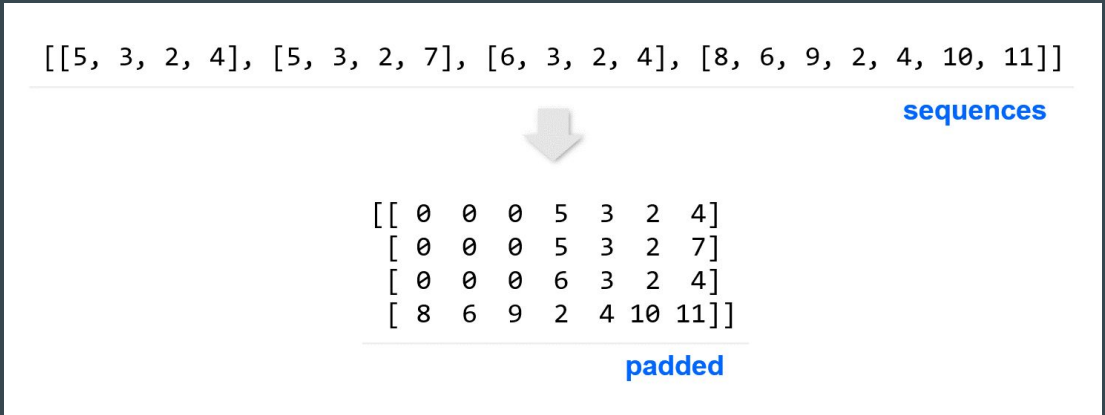



```
%%time
tokenizer = Tokenizer()
tokenizer.fit_on_texts(df_train.text)

vocab_size = len(tokenizer.word_index) + 1
print("Total words", vocab_size)
```

```
Total words 290419
```

사진 출처: https://codetorial.net/tensorflow/natural_language_processing_in_tensorflow_01.html

Pad Sequences



Sequential Model

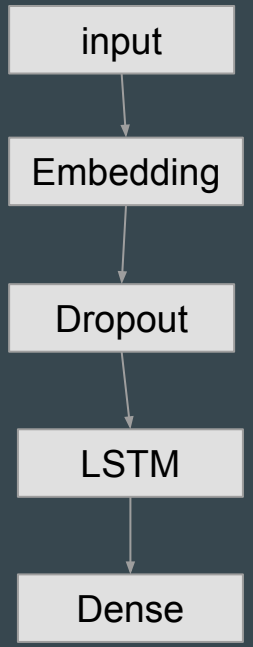
```
1  model = Sequential()
2  model.add(embedding_layer)
3  model.add(Dropout(0.5))
4  model.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2))
5  model.add(Dense(1, activation='sigmoid'))
6
7  model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 300, 300)          87125700
_________________________________________________________________
dropout (Dropout)            (None, 300, 300)          0
_________________________________________________________________
lstm (LSTM)                  (None, 100)               160400
_________________________________________________________________
dense (Dense)                (None, 1)                 101
=================================================================
Total params: 87,286,201
Trainable params: 160,501
Non-trainable params: 87,125,700
_________________________________________________________________
```

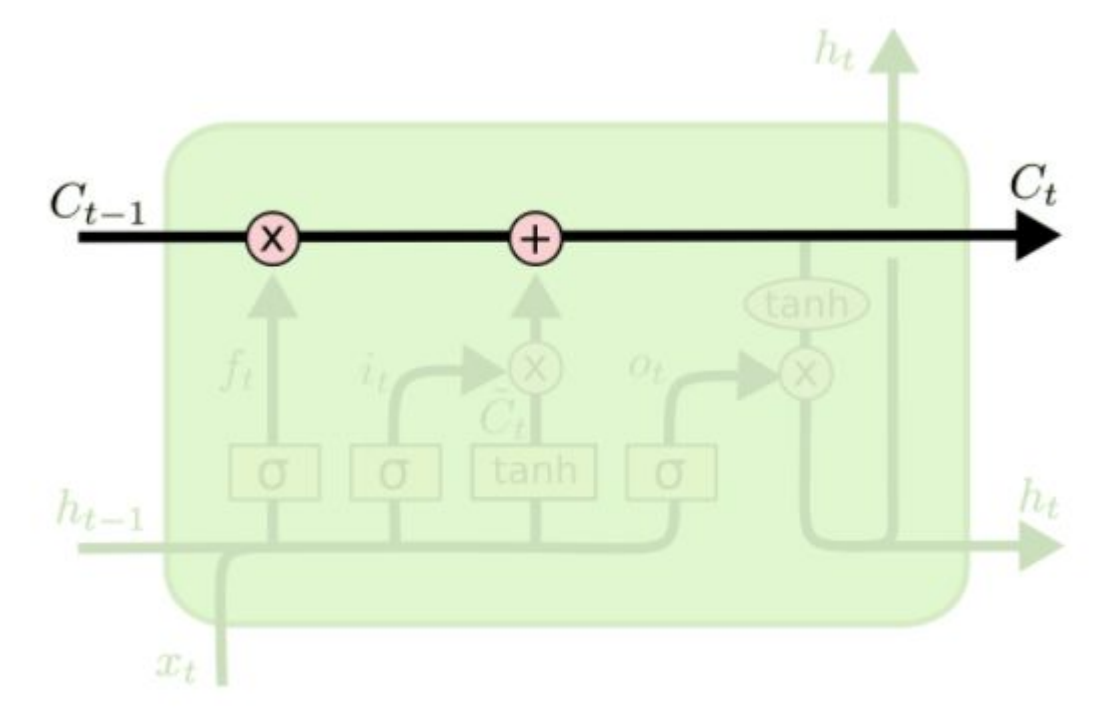
Sequential Model

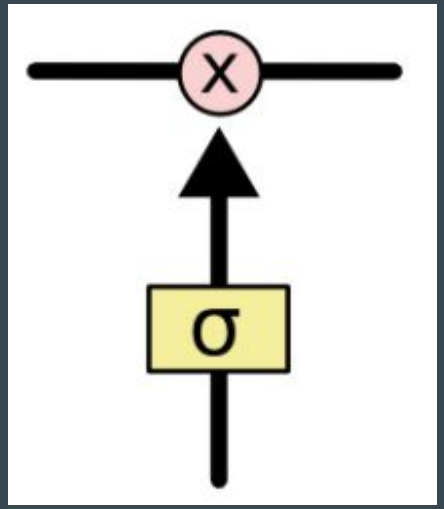
- model with single input, single output
- 4 layers

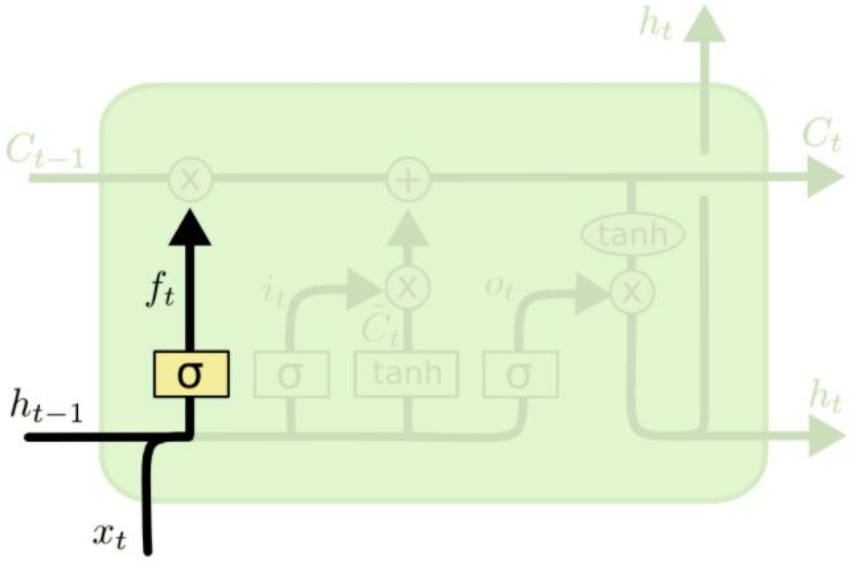
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 300, 300)          87125700
_________________________________________________________________
dropout (Dropout)            (None, 300, 300)          0
_________________________________________________________________
lstm (LSTM)                  (None, 100)               160400
_________________________________________________________________
dense (Dense)                (None, 1)                 101
=================================================================
Total params: 87,286,201
Trainable params: 160,501
Non-trainable params: 87,125,700
_________________________________________________________________
```



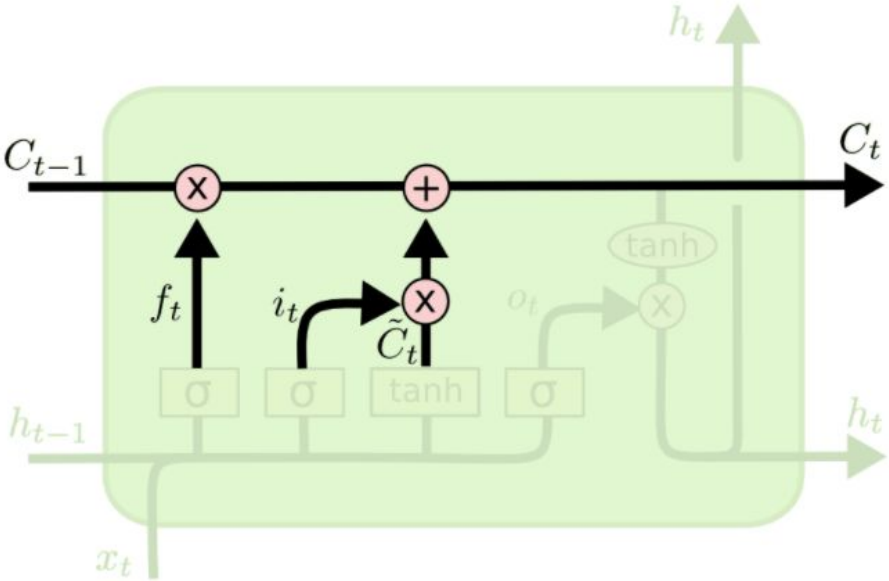
LSTM



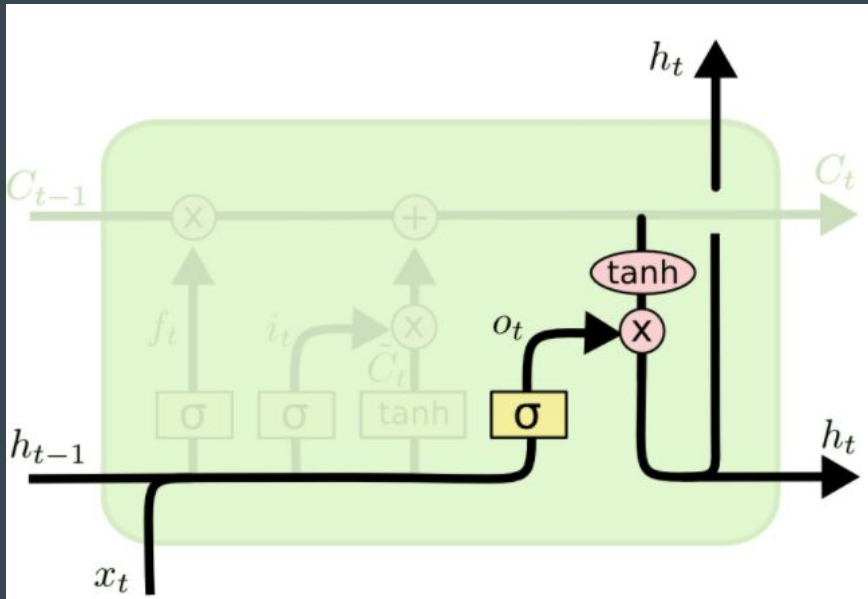




$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

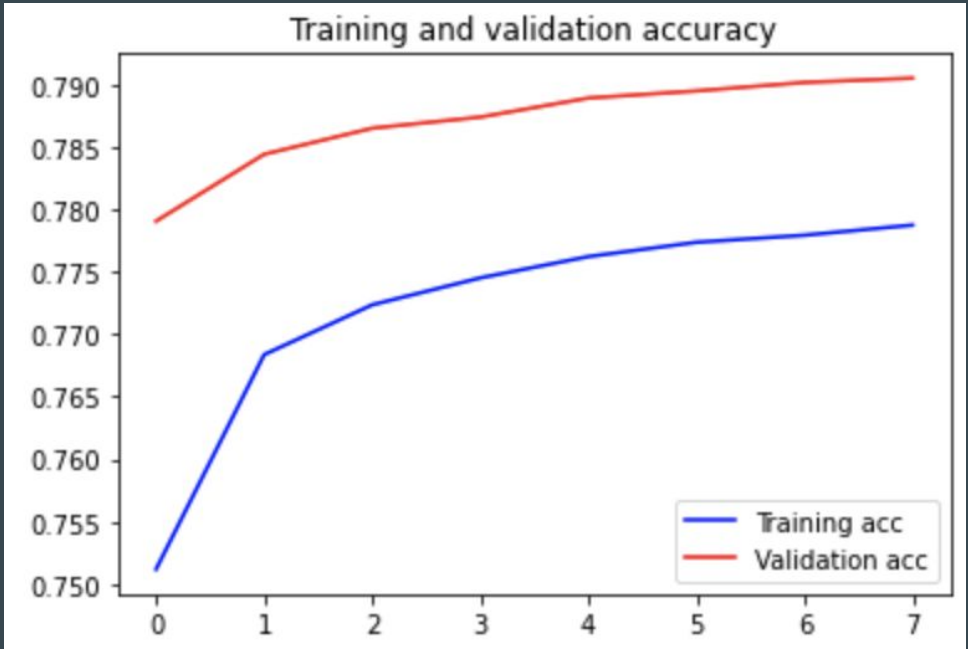
$$h_t = o_t * \tanh\left(C_t\right)$$

```
%%time
history = model.fit(x_train, y_train,
                    batch_size=BATCH_SIZE,
                    epochs=EPOCHS,
                    validation_split=0.1,
                    verbose=1,
                    callbacks=callbacks)
```

```
2020-12-05 07:08:03,346 : WARNING : Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy,lr

1125/1125 [==============================] - 8415s 7s/step - loss: 0.4634 - accuracy: 0.7774 - val_loss: 0.4478 - val_accuracy: 0.7895
Epoch 7/8
1125/1125 [==============================] - ETA: 0s - loss: 0.4622 - accuracy: 0.7779WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy,lr

2020-12-05 09:27:07,874 : WARNING : Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy,lr

1125/1125 [==============================] - 8336s 7s/step - loss: 0.4622 - accuracy: 0.7779 - val_loss: 0.4468 - val_accuracy: 0.7901
Epoch 8/8
1125/1125 [==============================] - ETA: 0s - loss: 0.4609 - accuracy: 0.7787WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy,lr

2020-12-05 11:51:28,396 : WARNING : Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy,lr

1125/1125 [==============================] - 8652s 8s/step - loss: 0.4609 - accuracy: 0.7787 - val_loss: 0.4452 - val_accuracy: 0.7905
CPU times: user 1d 18h 20s, sys: 18h 8min 45s, total: 2d 12h 9min 5s
Wall time: 20h 37min 52s
```

evaluation

```
313/313 [==============================] - 558s 2s/step - loss: 0.4441 - accuracy: 0.7913

ACCURACY: 0.791265606880188
LOSS: 0.4441477954387665
```



prediction

```
1 predict("I love the music")
```

```
{'label': 'POSITIVE',
 'score': 0.9617032408714294,
 'elapsed_time': 0.42303895950317383}
```

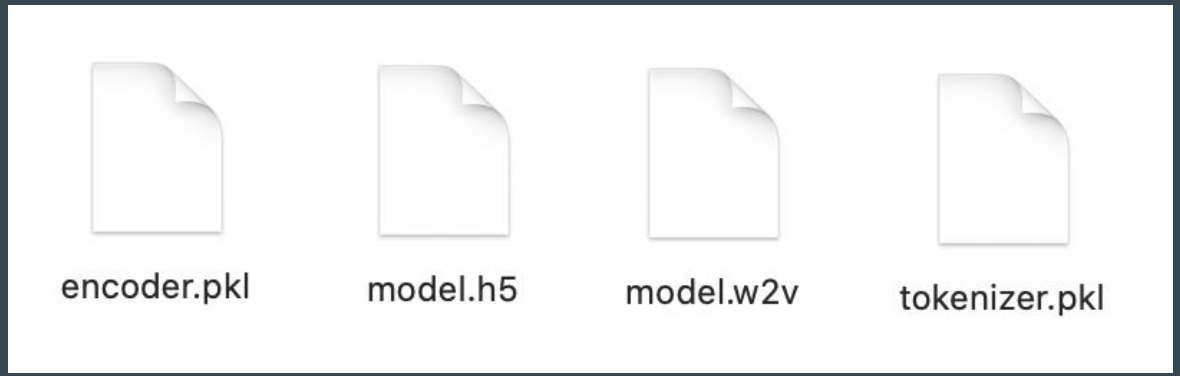
```
1 predict("I hate the rain")
```

```
{'label': 'NEGATIVE',
 'score': 0.010683596134185791,
 'elapsed_time': 0.108016967773437 5}
```

Saving Models

```
1  model.save(KERAS_MODEL)
2  w2v_model.save(WORD2VEC_MODEL)
3  pickle.dump(tokenizer, open(TOKENIZER_MODEL, "wb"), protocol=0)
4  pickle.dump(encoder, open(ENCODER_MODEL, "wb"), protocol=0)
```

```
2020-12-05 12:18:03,789 : INFO : saving Word2Vec object under model.w2v, separately None
2020-12-05 12:18:03,805 : INFO : not storing attribute vectors_norm
2020-12-05 12:18:03,807 : INFO : not storing attribute cum_table
2020-12-05 12:18:05,696 : INFO : saved model.w2v
```



comment_scraper.py

python comment_scraper.py [youtube url]
e.g. python comment_scraper.py https://www.youtube.com/watch?v=9bxc9hbwkkw

Username	Comment
Yellow sea	How many people are listening to this song in December 2020 ?♥♥♥♥♥♥♥♥♥ have a great day.
nisar ahmed	Who is listening in November 2020.? Hit like.. 💙
Huvo Kezo	Who's listening in November 2020?like💖
Kadek Yulia TV	Who's listening in August 2020 ? Like 💜
Lirik KIPIDAP	still listening this in 2016 ,
Cardi Bozzy	Who is listening in November 2020..
Dimas FilanOfficial	Who's Listening In September 2020? I swear this song I cry until I cover this song :(
Angela Ennin	This song never gets old... Its December 2020 and God has raised as up in this pandemic
X Triumph	who is living a heart for this song in December 2020 (today i jay kapster live a heart for this touching song on the 4th of December 2020)
R K	I lost my dad to the Covid 19 four days ago. Despite I'm in my mid 30s I was still daddy's girl, I can't imagine life without him. I kept hold o
khánh nguyễn	Song: You Raise Me Up LYRICS: When I am down, and, oh, my soul, so weary
Sunita Rani	Who is listening to this masterpiece in 2020?
Purple Truths	I lost my MOM - i am 40 and still not married - Dear God - Please help me to get over grief and find a good partner
Bernice Mkhabela	I lost a friend due to COVID-19.
	The good and innocent times 😭

app.py

app.py

```
app = Flask(__name__)
model = keras.models.load_model('model.h5')
tokenizer = pickle.load(open('tokenizer.pkl', 'rb'))
```

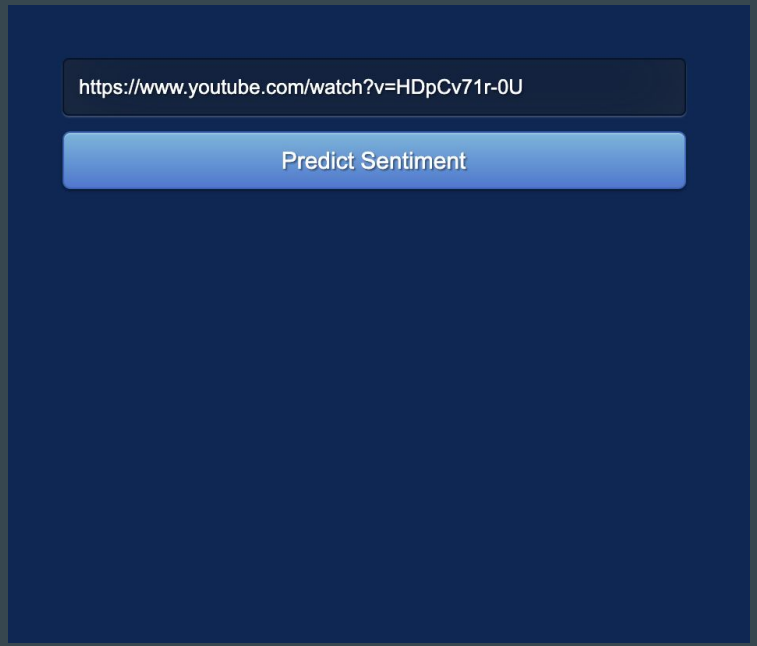
```
def predict_sentiment(text, include_neutral=True):
    start_at = time.time()
    # Tokenize text
    x_test = pad_sequences(tokenizer.texts_to_sequences([text]), maxlen=SEQUENCE_LENGTH)
    # Predict
    score = model.predict([x_test])[0]
    # Decode sentiment
    label = decode_sentiment(score, include_neutral=include_neutral)

    return {"label": label, "score": float(score),
       "elapsed_time": time.time()-start_at}
```

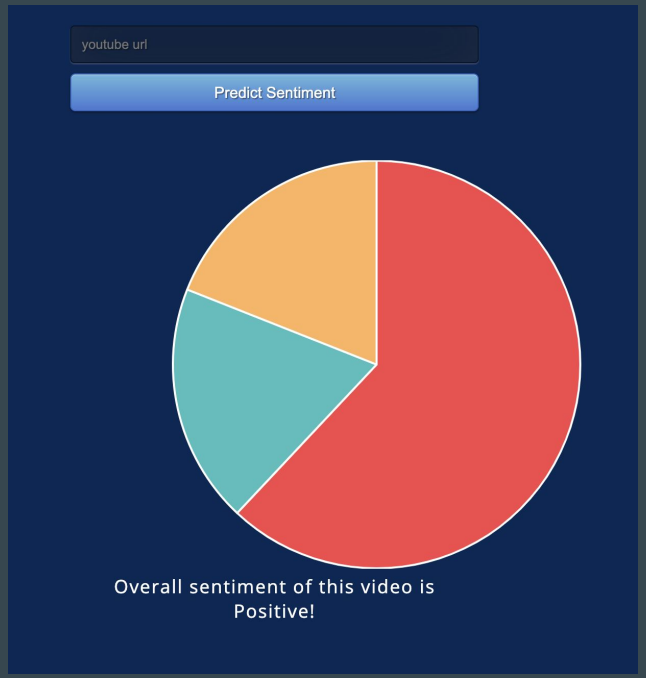
app.py

```
for i in range(len(df['Comment'])):
    comment = df['Comment'][i]

    prediction = predict_sentiment(comment)
    if prediction['label'] == POSITIVE:
        positives += 1
    elif prediction['label'] == NEUTRAL:
        neutrals += 1
    elif prediction['label'] == NEGATIVE:
        negatives += 1
    else:
        print("DEBUG : UNEXPECTED LABEL")
    print(comment, prediction['label'])
    storage.append((comment, prediction['label'], prediction['score']))

labels = 'Positive', 'Neutral', 'Negative'
sizes = [positives, neutrals, negatives]
colors = colors = ["#F7464A", "#46BFBD", "#FDB45C"]
```

app.py

Flask
web development,
one drop at a time





DEMO

감사합니다

참고자료

- Kaggle: Twitter Sentiment Analysis
  - https://www.kaggle.com/paoloripamonti/twitter-sentiment-analysis
- Understanding LSTM Networks
  - https://colah.github.io/posts/2015-08-Understanding-LSTMs/
- Kaggle: NLP beginner text classification using LSTM
  - https://www.kaggle.com/arunrk7/nlp-beginner-text-classification-using-lstm
- Medium: Word2Vec Explained
  - https://medium.com/datadriveninvestor/word2vec-skip-gram-model-explained-383fa6ddc4ae
- Github: flask-salary-predictor
  - https://github.com/vyashemang/flask-salary-predictor