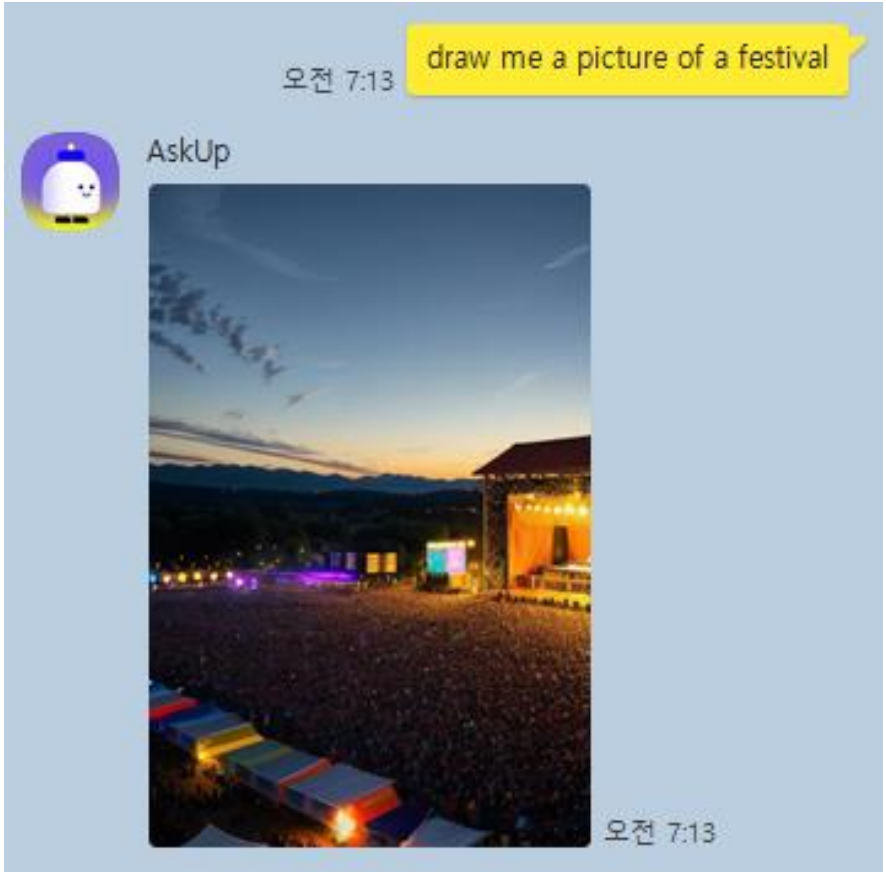
선도인재양성 중급

(11주차)

AI + X
Advanced Project



AskUp

카톡 친추

누가 그렸을까?

이것은?

✓ Big Picture

- 4/28(9강) AI MVP 개발, GPT-4
- 5/5 어린이날
- 5/12(10강) AI Service 경험 평가
- 5/19(11강) 미프 강의 (전처리, LSTM 개념)
- 5/26(12강) 과제 Checkup (15%) “장표, 발표 없음” (KT 코치 교수님 방문)
- 6/2(13강) 미프 강의 (LSTM 모델)
- 6/9(14강) 최종 과제 피드백
- 6/16(15강) 기말 프로젝트 발표 (30%) → 특허, 논문, 어워드

평가지표

Depth of Learning

잠재적 사용자로부터

얼마나 깊이 있게 배웠는가?

[옵션]

빅데이터, AI 경진대회

AWS, KT

Lean 철학

The process now
is continuous engagement

관여, 몰입

– 소비자를 참여시켜 끊임없이 피드백을 받음

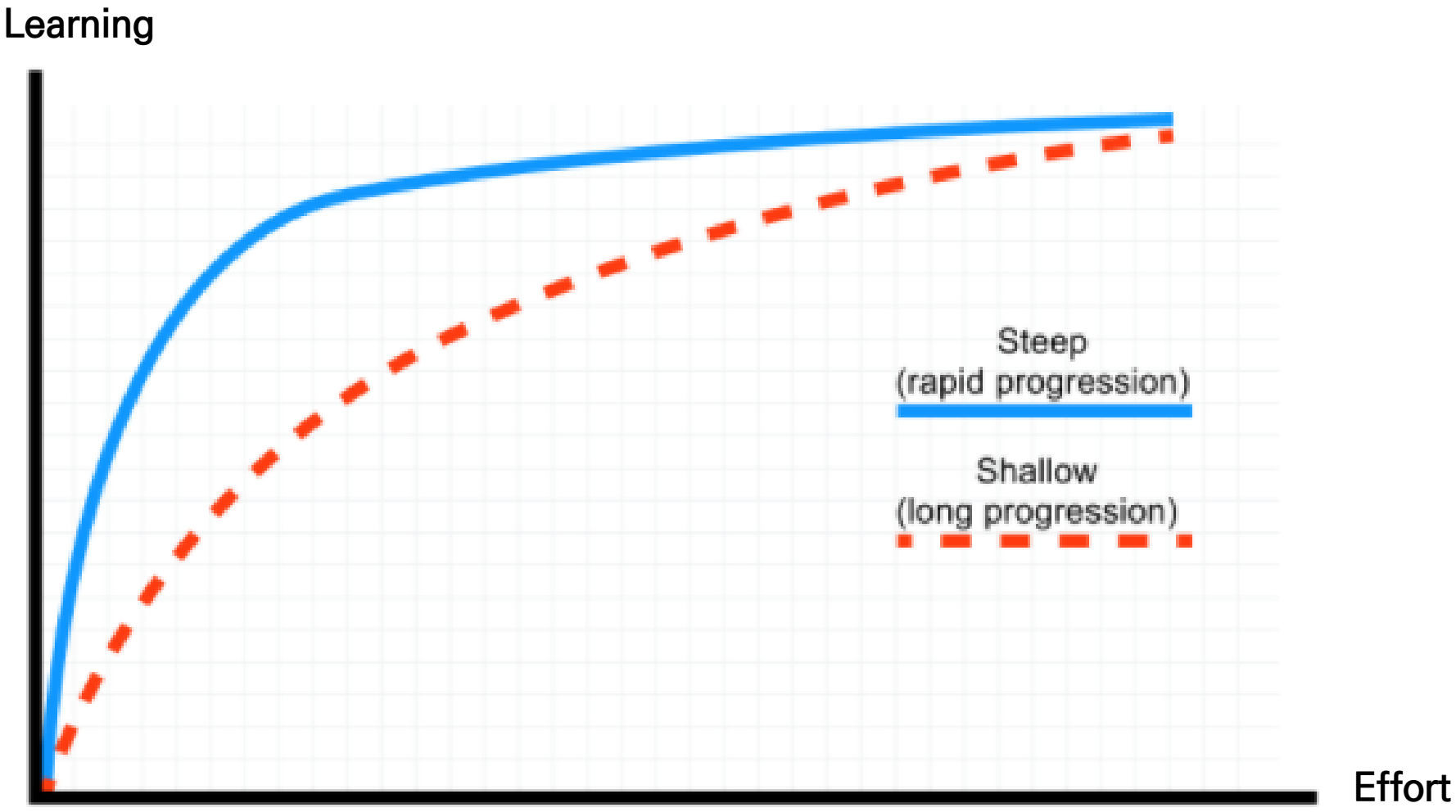
Lean =

Cost Effective Design & Evaluation

가성비 설계 및 평가

최소한의 투자로 최대한을 배운다!

Lean adjective not containing any fat (the waste)


Learning
Steep
(rapid progression)
Shallow
(long progression)
Effort

The more evidence you have,

The higher the probability of

success of your product

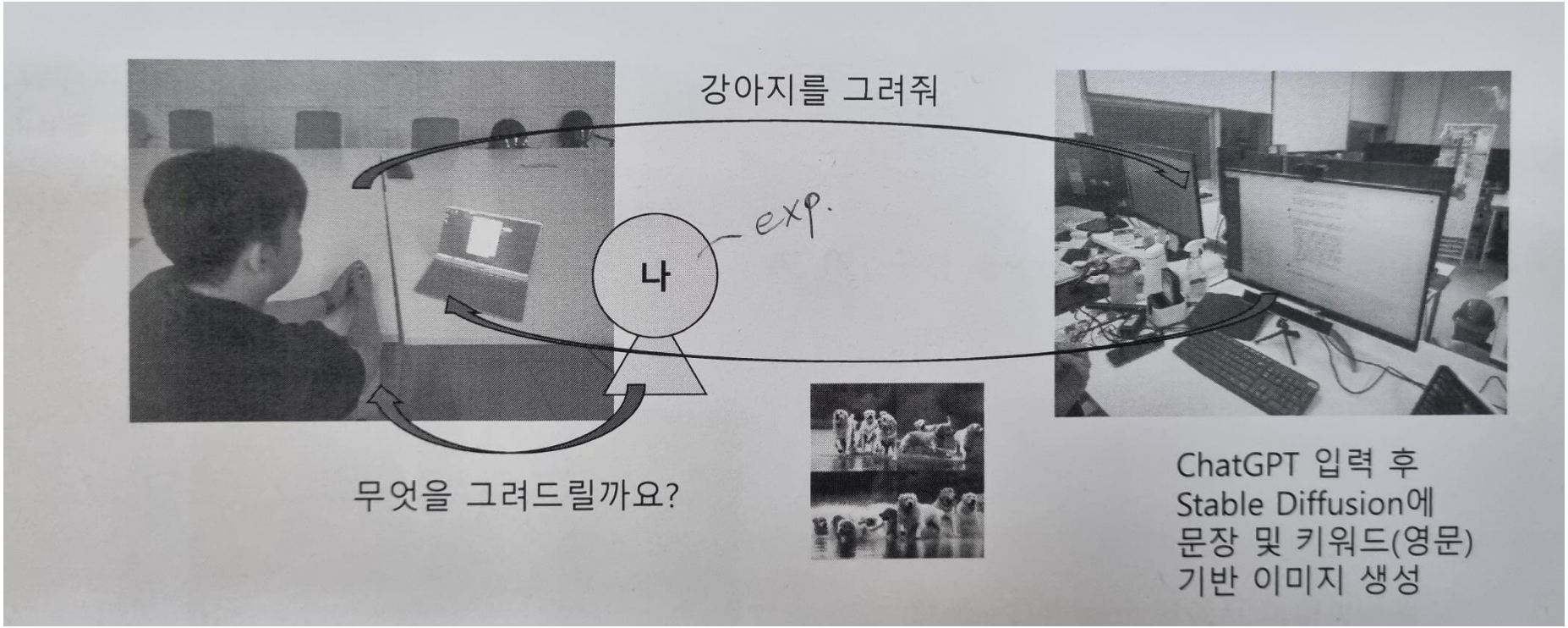
제품 및 서비스의 성공률을

높이기 위한 증거를 얻어야함!

Rapid
AI Service Evaluation

Cost Effective Evaluation

Wizard of Oz



Wizard of Oz Example

미니 프로젝트 해부

ML을 Deep을 하든 데이터부터 살펴보기

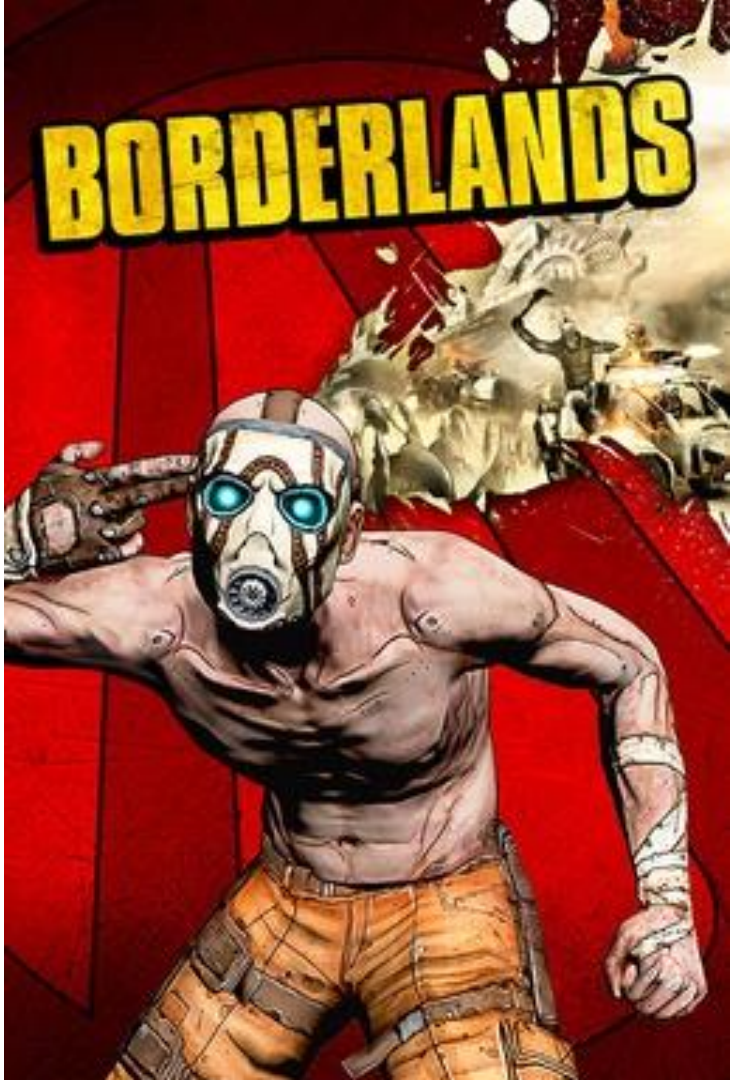
Attention to Detail!

```
[ ] data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 74682 entries, 0 to 74681
Data columns (total 4 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   Tweet_ID       74682 non-null  int64 
 1   Entity         74682 non-null  object
 2   Sentiment      74682 non-null  object
 3   Tweet_content  73996 non-null  object
dtypes: int64(1), object(3)
memory usage: 2.3+ MB
```

```
[ ] data.head()
```

	Tweet_ID	Entity	Sentiment	Tweet_content
0	2401	Borderlands	Positive	im getting on borderlands and i will murder yo...
1	2401	Borderlands	Positive	I am coming to the borders and I will kill you...
2	2401	Borderlands	Positive	im getting on borderlands and i will kill you ...
3	2401	Borderlands	Positive	im coming on borderlands and i will murder you...
4	2401	Borderlands	Positive	im getting on borderlands 2 and i will murder ...


BORDERLANDS

```
data.Entity.unique()
```

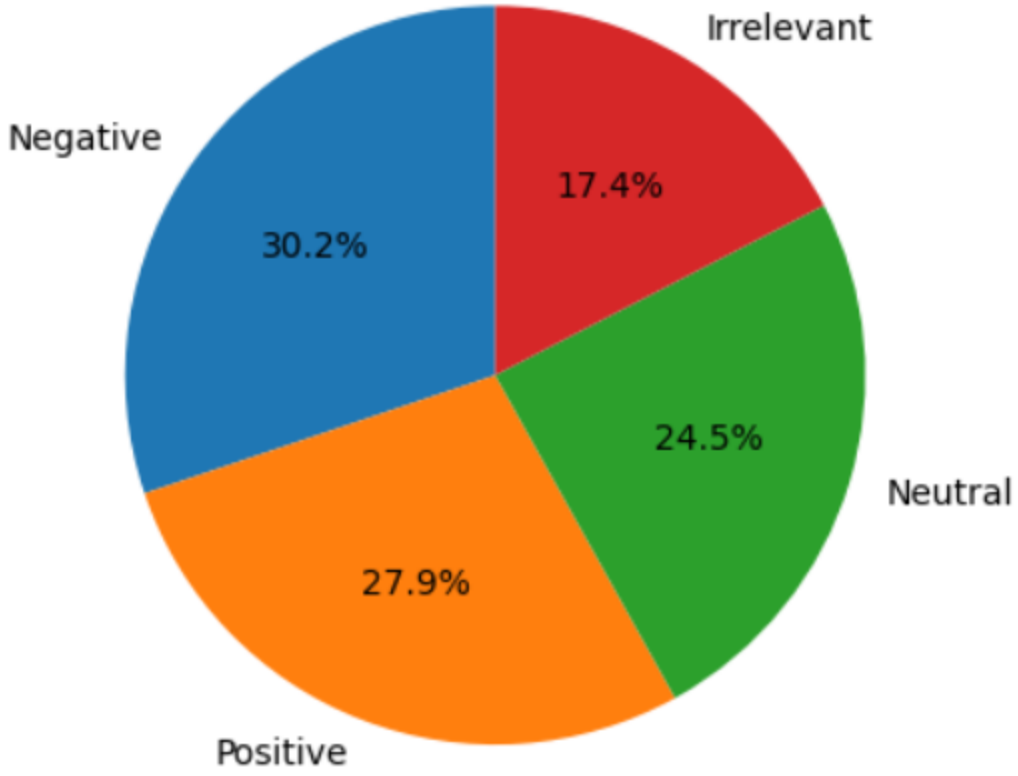
```
array(['Borderlands', 'CallOfDutyBlackopsColdWar', 'Amazon', 'Overwatch',
       'Xbox(Xseries)', 'NBA2K', 'Dota2', 'PlayStation5(PS5)',
       'WorldOfCraft', 'CS-GO', 'Google', 'AssassinsCreed', 'ApexLegends',
       'LeagueOfLegends', 'Fortnite', 'Microsoft', 'Hearthstone',
       'Battlefield', 'PlayerUnknownsBattlegrounds(PUBG)', 'Verizon',
       'HomeDepot', 'FIFA', 'RedDeadRedemption(RDR)', 'CallOfDuty',
       'TomClancysRainbowSix', 'Facebook', 'GrandTheftAuto(GTA)',
       'MaddenNFL', 'johnson&johnson', 'Cyberpunk2077',
       'TomClancysGhostRecon', 'Nvidia'], dtype=object)
```

```
[ ] data.Sentiment.unique()
```

```
array(['Positive', 'Neutral', 'Negative', 'Irrelevant'], dtype=object)
```

```
[ ] plt.pie(data.Sentiment.value_counts(),labels=['Negative','Positive','Neutral','Irrelevant'],autopct='%1.1f%%',startangle=90)
```

```
([<matplotlib.patches.Wedge at 0x7ffae44318a0>,
  <matplotlib.patches.Wedge at 0x7ffae4431e10>,
  <matplotlib.patches.Wedge at 0x7ffae20f8b80>,
  <matplotlib.patches.Wedge at 0x7ffae20f9210>],
 [Text(-0.8936408809046303, 0.6414093669225578, 'Negative'),
  Text(-0.39649770564254117, -1.0260553442286633, 'Positive'),
  Text(1.0532293937999033, -0.3173449921392938, 'Neutral'),
  Text(0.5716146417662045, 0.9398173765782871, 'Irrelevant')],
 [Text(-0.4874404804934347, 0.3498596546850315, '30.2%'),
```


Irrelevant
17.4%
Negative
30.2%
24.5%
Neutral
27.9%
Positive

```
[ ] data['Sentiment'].value_counts()
```

```
Negative      22542
Positive      20832
Neutral       18318
Irrelevant    12990
Name: Sentiment, dtype: int64
```

```
[ ] data.duplicated().sum()
```

```
2700
```

```
[ ] data.drop_duplicates(inplace=True)
    data.duplicated().sum()
```

```
0
```

```
[ ] data.isnull().sum()
```

```
Tweet_ID          0
Entity            0
Sentiment         0
Tweet_content   326
dtype: int64
```

```
data.dropna(axis=0, inplace=True)
data.isnull().sum()
```

```
Tweet_ID          0
Entity            0
Sentiment         0
Tweet_content     0
dtype: int64
```

```
[ ] data.reset_index(inplace=True)
```

```
[ ] data.shape
```

```
(71656, 5)
```

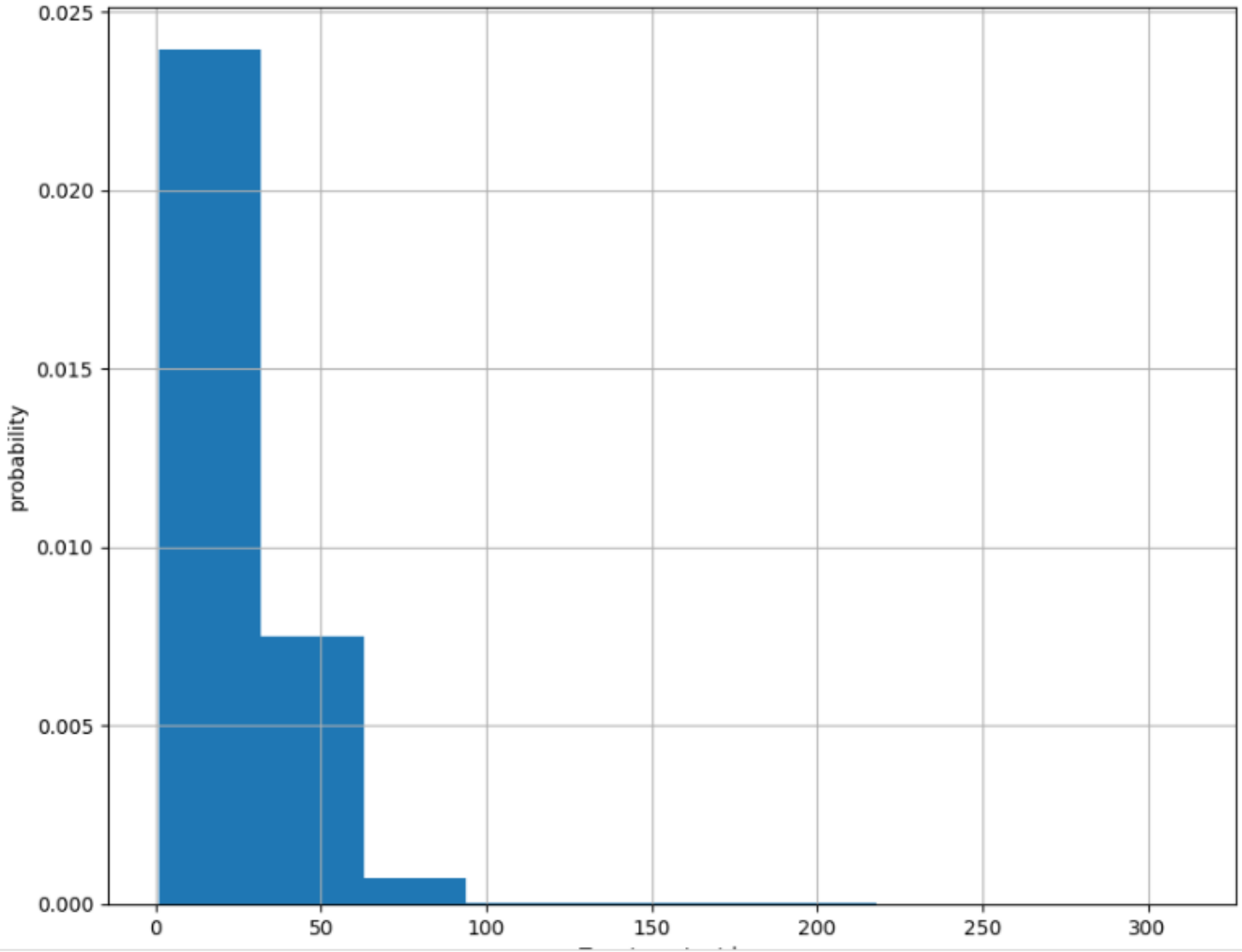
```
[ ] data.head()
```

	index	Tweet_ID	Entity	Sentiment	Tweet_content
0	0	2401	Borderlands	Positive	im getting on borderlands and i will murder yo...
1	1	2401	Borderlands	Positive	I am coming to the borders and I will kill you...
2	2	2401	Borderlands	Positive	im getting on borderlands and i will kill you ...
3	3	2401	Borderlands	Positive	im coming on borderlands and i will murder you...
4	4	2401	Borderlands	Positive	im getting on borderlands 2 and i will murder ...

```
[ ]  replace_list = {r"i'm": 'i am',
                     r"'re": ' are',
                     r"let' s": 'let us',
                     r"'s":  ' is',
                     r"'ve": ' have',
                     r"can't": 'can not',
                     r"cannot": 'can not',
                     r"shan' t": 'shall not',
                     r"n't": ' not',
                     r"'d": ' would',
                     r"'ll": ' will',
                     r"'scuse": 'excuse',
                     ',': ' ,',
                     '.': ' .',
                     '!': ' !',
                     '?': ' ?',
                     '₩s+': ' '}
     def clean_text(text):
         text = text.lower()
         for s in replace_list:
             text = text.replace(s, replace_list[s])
         text = ' '.join(text.split()) # removing whitespace
         return text
```

```
Tweet_content_len = x.apply(lambda p: len(p.split(' ')))
max_Tweet_content_len = Tweet_content_len.max()
print('max Tweet_content len: {0}'.format(max_Tweet_content_len))
plt.figure(figsize = (10, 8))
plt.hist(Tweet_content_len, density = True)
plt.xlabel('Tweet_content len')
plt.ylabel('probability')
plt.grid()
```

```
max Tweet_content len: 311
```



```
[ ]  import re
     REPLACE_WITH_SPACE = re.compile("(@)")
     SPACE = " "
     from nltk.corpus import stopwords
     english_stop_words = stopwords.words('english')
     from nltk.stem.porter import PorterStemmer
     #1
     def reviews(reviews):
         reviews = [REPLACE_WITH_SPACE.sub(SPACE, line.lower()) for line in reviews]
         return reviews
     #2
     def remove_stop_words(corpus):
         removed_stop_words = []
         for review in corpus:
             removed_stop_words.append(
                 ' '.join([word for word in review.split()  if word not in english_stop_words]))
         return removed_stop_words
     #3
     def get_stemmed_text(corpus):
         stemmer = PorterStemmer()

         return [' '.join([stemmer.stem(word) for word in review.split()]) for review in corpus]
```

```
[ ]  import re
     REPLACE_WITH_SPACE = re.compile("(@)")
     SPACE = " "
     from nltk.corpus import stopwords
     english_stop_words = stopwords.words('english')
     from nltk.stem.porter import PorterStemmer
     #1
     def reviews(reviews):
         reviews = [REPLACE_WITH_SPACE.sub(SPACE, line.lower()) for line in reviews]
         return reviews
     #2
     def remove_stop_words(corpus):
         removed_stop_words = []
         for review in corpus:
             removed_stop_words.append(
                 ' '.join([word for word in review.split()  if word not in english_stop_words]))
         return removed_stop_words
     #3
     def get_stemmed_text(corpus):
         stemmer = PorterStemmer()

         return [' '.join([stemmer.stem(word) for word in review.split()]) for review in corpus]
```

불용어 제거

어간(원형) 축소

Stopwords

불용어

큰 의미가 없는 단어 제거

Stopwords Example

```
example = "Family is not an important thing. It's everything."
stop_words = set(stopwords.words('english'))

word_tokens = word_tokenize(example)

result = []
for word in word_tokens:
    if word not in stop_words:
        result.append(word)

print('불용어 제거 전 :',word_tokens)
print('불용어 제거 후 :',result)
```

```
불용어 제거 전 : ['Family', 'is', 'not', 'an', 'important', 'thing', '.', 'It', "'s", 'everything', '.']
불용어 제거 후 : ['Family', 'important', 'thing', '.', 'It', "'s", 'everything', '.']
```

Stopwords (불용어)

- I, my, me, over, 조사, 접미사

- 자주 등장하지만 실제 의미 분석을 하는데 거의 기여하는 바가 없는 경우

- NLTK 패키지에서 100여개 이상의 영단어들을 불용어로 선정

보편적인 한국어 불용어 리스트

https://www.ranks.nl/stopwords/korean

Korean Stopwords

Korean Stopwords

아	어찌됏든	하기보다는
휴	그위에	차라리
아이구	게다가	하는 편이 낫다
아이쿠	점에서 보아	흐흐
아이고	비추어 보아	놀라다
어	고려하면	상대적으로 말하
나	하게될것이다	자면
우리	일것이다	마치
저희	비교적	아니라면
따라	좀	쉿
의해	보다더	그렇지 않으면
을	비하면	그렇지 않다면
를	시키다	안 그러면
에	하게하다	아니었다면
의	할만하다	하든지
가	의해서	아니면
으로	연이서	이라면
로	이어서	좋아
에게	잇따라	알았어
뿐이다	뒤따라	하는것도
의거하여	뒤이어	그만이다
근거하여	결국	어쩔수 없다
입각하여	의지하여	하나
기준으로	기대여	일
예하면	통하여	일반적으로
예를 들면	자마자	일단
예를 들자면	더욱더	한켠으로는
저	불구하고	오자마자
소인	얼마든지	이렇게되면
소생	마음대로	이와같다면
저희	주저하지 않고	전부
지말고	곧	한마디
하지마	즉시	한항목
하지마라	바로	근거로
다른	당장	하기에
물론	하자마자	아울러

Stemming (원형 축소)
어간

- NLP의 텍스트 전처리 기법
- 접사(Suffix)을 제거해 단어를 원형(어간)으로 축소함
- running, runs, ran → 어간: run
- Text을 정규화해서 텍스트 분류, 문서 군집화, 정보 검색 등에 활용
- e.g., Porter Stemming Algorithm

```
[ ]  stemmed_reviews_tweet[20]
```

```
'first borderland session long time actual enjoy realli satisfi combat experi . got rather good kill'
```

```
[ ]  max_words = 8000

     tokenizer = Tokenizer(
         num_words = max_words,
         filters = '"#$%&()*+-/:;<=>@[₩]^_`{|}~'
     )
     tokenizer.fit_on_texts(stemmed_reviews_tweet)
     x = tokenizer.texts_to_sequences(stemmed_reviews_tweet)
     x = pad_sequences(x, maxlen = 300)
```

// updates the internal vocabulary of texts

// turns the data into sequences of integers

```
[ ]  label_tokenizer = Tokenizer()
     label_tokenizer.fit_on_texts(y)
```

Keras ‘Tokenizer’ class!

LSTM

Long Short-Term Memory

저는 카투사로 군을 다녀왔습니다 . . .

자유시간 많았구요.

그래도 전방이라 훈련 많이 댕겼구요.

(블라 블라)

(블라 블라)

(블라 블라)

(블라 블라)

(블라 블라)

(블라 블라)

(블라 블라)

(블라 블라)

(블라 블라) Zzzz

(블라 블라)

(블라 블라)

(블라 블라)

그래서 저는 Low Quality 영어를 합니다.

저질

(블라 블라)

(블라 블라)

(블라 블라)

그래서 저는 Low Quality  를 합니다.

저질

Recurrent

RNN (순환신경망)

입력값과 그 값을 사용하는 지점의 거리가

멀수록 학습 능력이 격감

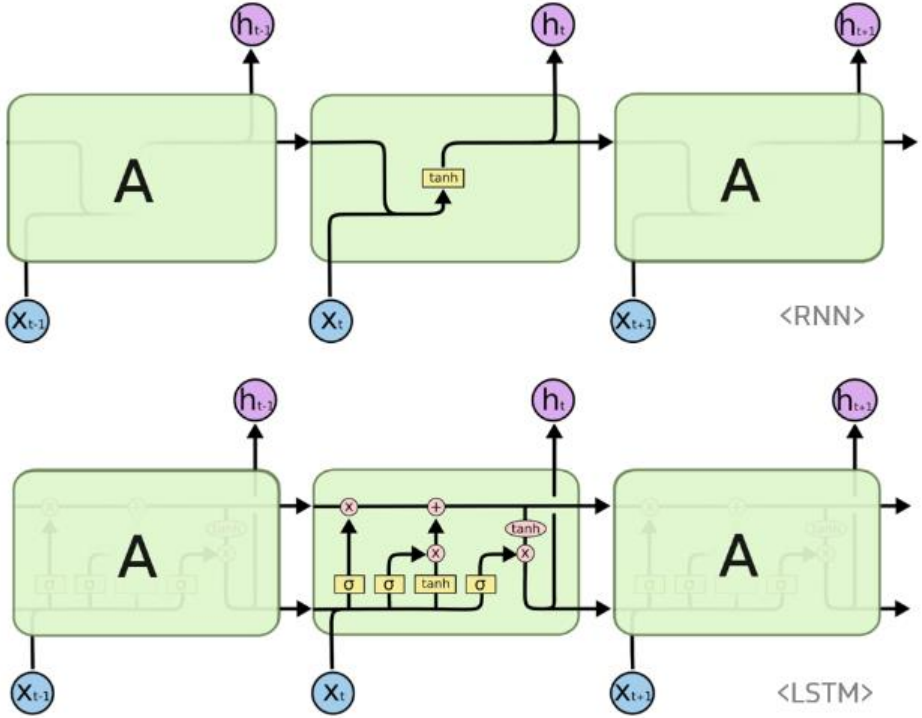
Gradient Vanishing Problem

기울기 소실 문제

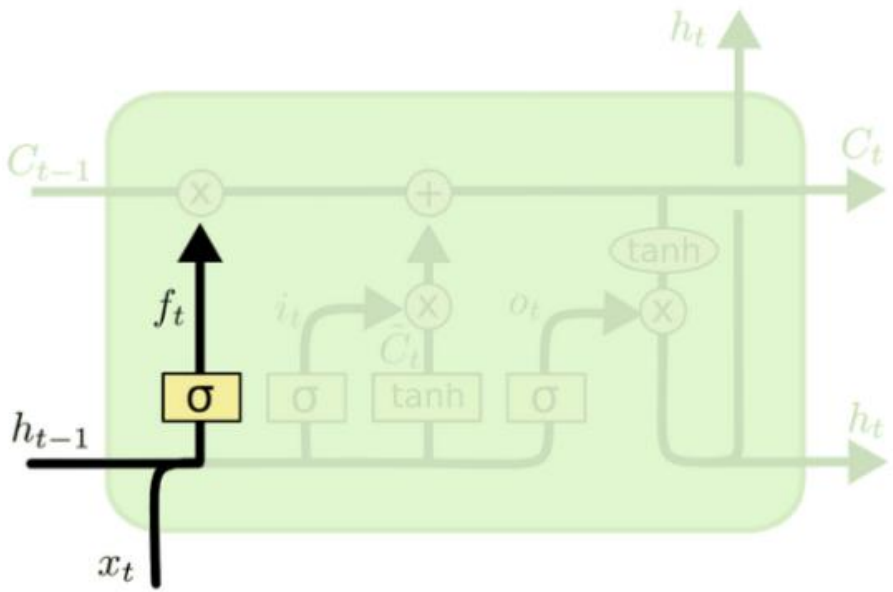
RNN은 역전파를 통해 학습하는데 이전 단어와 다음 단어 사이의 연결고리를 강화하기 위해 기울기 사용

기울기는 역전파 과정에서 곱셈 연산이 반복적으로 이루어지기 때문에 매단계마다 기울기 값이 곱해지면서 지수적으로 감소

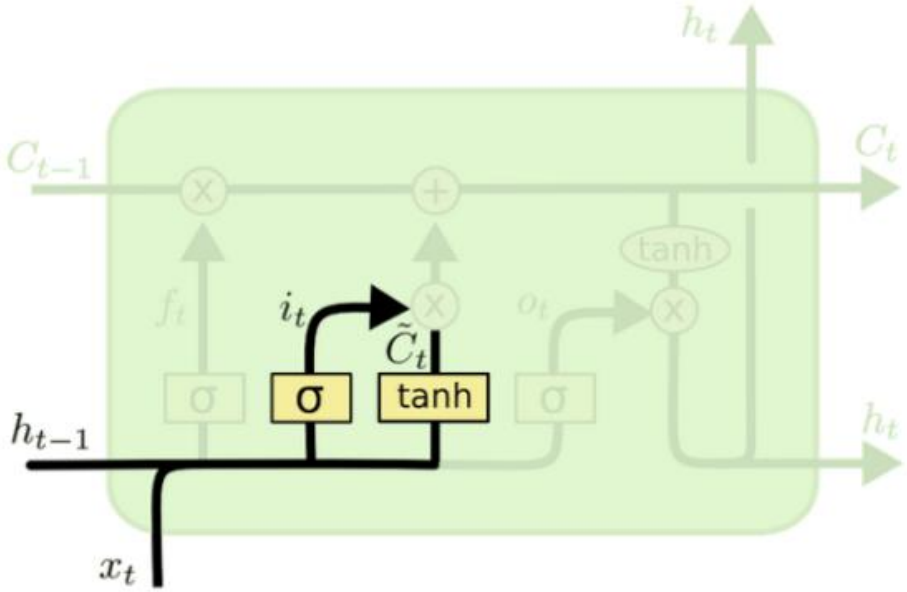
LSTM은 오랜 기간 정보를 저장할 수 있도록 함!



0~1 값을 곱해 과거 정보를 얼마나 기억할지 결정



<forget gate>



<input gate>

현재 정보를 얼마나 기억할지 결정하는 게이트

```
model_lstm = Sequential()
model_lstm.add(Embedding(input_dim = max_words, output_dim = 128, input_length = 300))
model_lstm.add(SpatialDropout1D(0.3))
model_lstm.add(LSTM(128, dropout = 0.3, recurrent_dropout = 0.3))
model_lstm.add(Dense(128, activation = 'relu'))
model_lstm.add(Dropout(0.3))
model_lstm.add(Dense(5, activation = 'softmax'))
model_lstm.compile(
    loss='sparse_categorical_crossentropy',
    optimizer='Adam',
    metrics=['accuracy']
)
```

모델 설계 및 학습 부분은 다음 시간에. . .

Group Discussion