

# Paper Review

이 한 준 교수

# 논문을 이해하기 위한 사전지식

- 감성분석(Sentiment Analysis)
  - Sentiment(감성)
    - 긍정(Positive), 부정(Negative), 중립(Neutral)과 같은 감성의 전반적인 경향
  - 감성분석은 텍스트가 어떤 감성을 띠고 있는지를 분류하는 것이 핵심
    - 긍정/부정, 긍정/중립/부정 등으로 분류
    - 예시
      - "이 영화 정말 재미있었어요!" → 긍정(Positive)
      - "진짜 지루하고 시간 낭비였어." → 부정(Negative)
      - "그냥 그랬다." → 중립(Neutral)

# 논문을 이해하기 위한 사전지식

- 감성분석(Sentiment Analysis)

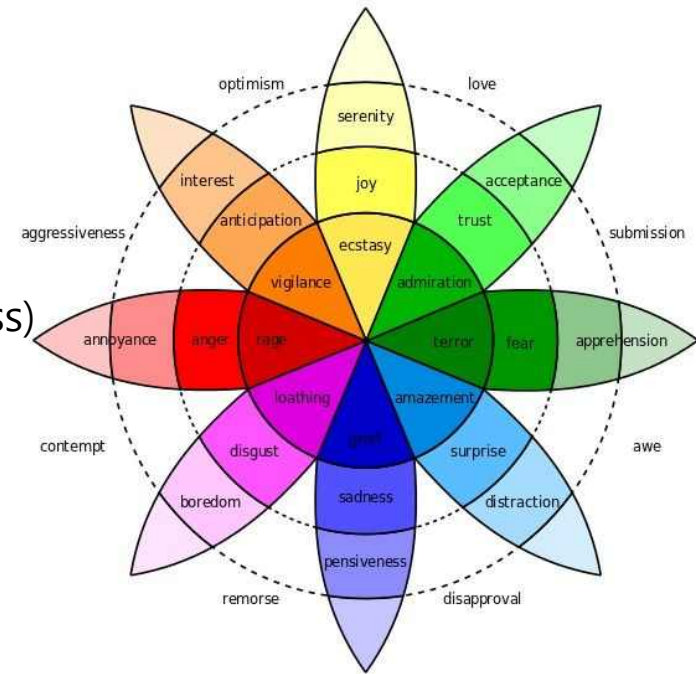
- 참고) Emotion(감정) vs. Sentiment(감성)

- 감정은 기쁨(Joy), 분노(Anger), 슬픔(Sadness), 두려움(Fear), 놀람(Surprise) 등 다양한 감정 상태

- 심리학의 감정 모델을 따름(예시: Plutchik)

- 예시

- "이 영화 최고야! 정말 행복해!" → 기쁨(Joy)
      - "너무 화나! 돈이 아깝다." → 분노(Anger)
      - "이 장면 너무 슬퍼서 눈물이 난다." → 슬픔(Sadness)
      - "이게 말이 돼? 믿을 수 없어!" → 놀람(Surprise)



# 논문을 이해하기 위한 사전지식

- 감성분석(Sentiment Analysis)
  - 감성을 컴퓨터에게 어떻게 학습시킬까?
    1. 감성 사전(Sentiment Lexicon) 기반 접근 방식
      - 미리 단어에 대한 점수를 부여하고 전체 문장의 감성을 계산
      - 예시) "좋다" (+1), "나쁘다" (-1), "최고" (+2), "별로" (-2)
    2. 머신러닝 기반 접근 방식
      - 긍정과 부정 텍스트를 지도학습을 통해 학습시킴

# 논문을 이해하기 위한 사전지식

- 감성분석(Sentiment Analysis)
  - 감성을 컴퓨터에게 어떻게 학습시킬까?
    1. 감성 사전(Sentiment Lexicon) 기반 접근 방식
      - 감성 사전 준비
        - SentiWordNet – 예) "good" adj. P=0.8, N=0.1
        - VADER
        - 한국어 감성 사전
      - 텍스트 전처리 (Tokenization, 불용어 제거 등)
      - 감성 단어 매칭 후 점수 합산
  - 감성 사전 기반으로 감성 측정시 어떤 문제가 있을까?

# 논문을 이해하기 위한 사전지식

- 감성분석(Sentiment Analysis)
  - 감성을 컴퓨터에게 어떻게 학습시킬까?
    1. 감성 사전(Sentiment Lexicon) 기반 접근 방식
      - 장점
        - 데이터 학습이 필요없음
        - 감성 사전의 품질에 따라 높은 정확도 기대 가능
      - 단점
        - 감성 사전에 없는 단어 분석 어려움
        - 문맥 반영이 어려움 – 예) 너무 좋지는 않다.
        - 도메인별 감성 특성 반영 어려움 – 예) 영화리뷰에서 “시간”

# 논문을 이해하기 위한 사전지식

- 감성분석(Sentiment Analysis)
  - 감성을 컴퓨터에게 어떻게 학습시킬까?
    2. 머신러닝 기반 접근 방식
      - 텍스트 전처리와 언어모델, 머신러닝 모델 사용
        - 주로 텍스트 임베딩 사용(TF-IDF, Word2Vec, BERT)
        - 머신러닝(딥러닝)으로 학습

# 논문을 이해하기 위한 사전지식

- 감성분석(Sentiment Analysis)
  - 감성을 컴퓨터에게 어떻게 학습시킬까?
    2. 머신러닝 기반 접근 방식
      - 장점
        - 감성 사전 방식보다 문맥과 도메인을 잘 반영하여 높은 성능 기대할 수 있음
      - 단점
        - 많은 양의 학습용 데이터 필요
        - 사람이 수작업으로 학습을 시켜야할 수도 있음



# Paper Critique

- Twitter mood predicts the stock market

검색결과 약 29,900개 (0.15초)

---

## Twitter mood predicts the stock market

[PDF] [arxiv.org](#)

[J Bollen](#), [H Mao](#), [X Zeng](#) - Journal of computational science, 2011 - Elsevier

... **mood** states derived from large-scale **Twitter** ... **Twitter** feeds by two **mood** tracking tools, namely OpinionFinder that measures positive vs. negative **mood** and Google-Profile of **Mood** ...

☆ 저장 99 인용 7493회 인용 관련 학술자료 전체 30개의 버전

JUST A NORMAL DAY AT THE NATION'S MOST IMPORTANT FINANCIAL INSTITUTION...



# 이름 비슷하다고 엉뚱한 종목 투자... '묻지마' 테마주

동아일보 | 업데이트 2019-11-26 11:40

## ○ 회사 이름 비슷하다고 엉뚱한 회사에 투자자 몰려



“뇌전증 신약 ‘엑스코프리’(성분명 세노바메이트)를 개발한 회사는 SK바이오랜드가 아니라 SK 바이오팜 아닌가요?” “그런 거 따질 시간에도 주식은 계속 오르니 일단 사고 봅시다.”

SK그룹 자회사 SK바이오팜의 독자 개발 신약이 미국 식품의약국(FDA)의 공식 판매 허가를 받았다는 발표가 났던 22일, SK바이오팜과 사명(社名)이 비슷한 코스닥 상장사 SK바이오랜드의 주가가 가격제한폭(30%)까지 오르자 주식투자 관련 인터넷 카페에는 혼란스러워하는 투자자들의 질문이 이어졌다. SK바이오랜드의 주가는 다음 거래일인 25일에도 투자자들의 ‘묻지마 투자’ 행진이 이어지며 2거래일 연속 상한가를 기록했다.

# Introduction

- Limitation of predicting the stock market
  - Early research – 주식은 예측 불가!
    - Random walk theory
    - Efficient Market Hypothesis(EMH)
      - Stock market prices are driven by new information(NEWS) rather than present and past prices
  - Later research – 인간의 감정이 의사 결정에 영향
    - Socio-nomic Theory of Finance(STF)
    - Behavioral economics
    - Behavioral finance
- News may be unpredictable,  
but indicators can be extracted from online social media

# Approach

- Data collection
  - Feb. 28. '08 – Dec. 19. '08
  - 9,853,498 tweets by 2.7M users
- Data Filtering
  - Tweet selection
    - "I feel", "I'm feeling", "I don't feel", "I am", ...

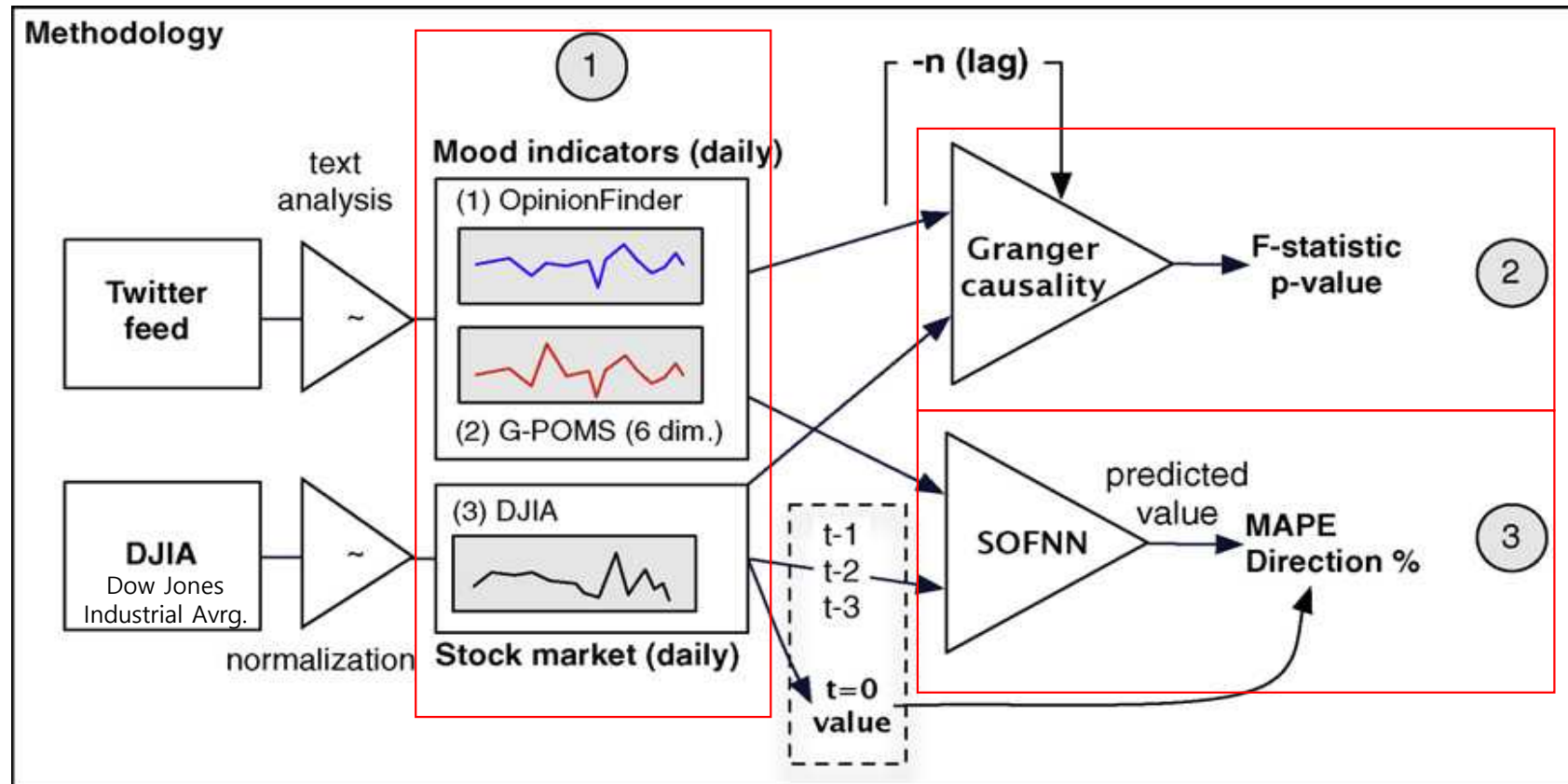
# Approach

- Data gathering Tool
  - Opinion Finder(<https://mpqa.cs.pitt.edu/opinionfinder/> 다운로드 후 실행. JDK 필요)
    - 사전을 기반으로 문장 단위 긍정/부정 판단 가능
  - GPOMS
    - 인간 감정을 6가지 범주로 분류
      - Calm, Alert, Sure, Vital, Kind and Happy
    - 사전 단어들과의 매칭을 통하여 특정 감정의 점수를 계산

# Approach

Granger causality(그랜저 인과성)

시계열 데이터에서 한 변수(X)가 다른 변수(Y)의 미래 값을 예측하는 데 유의미한 정보를 제공하는지를 검정하는 방법입니다. 단순한 상관관계와 달리, 시간적 선후 관계를 고려하여 원인과 결과의 관계를 분석하는 데 사용

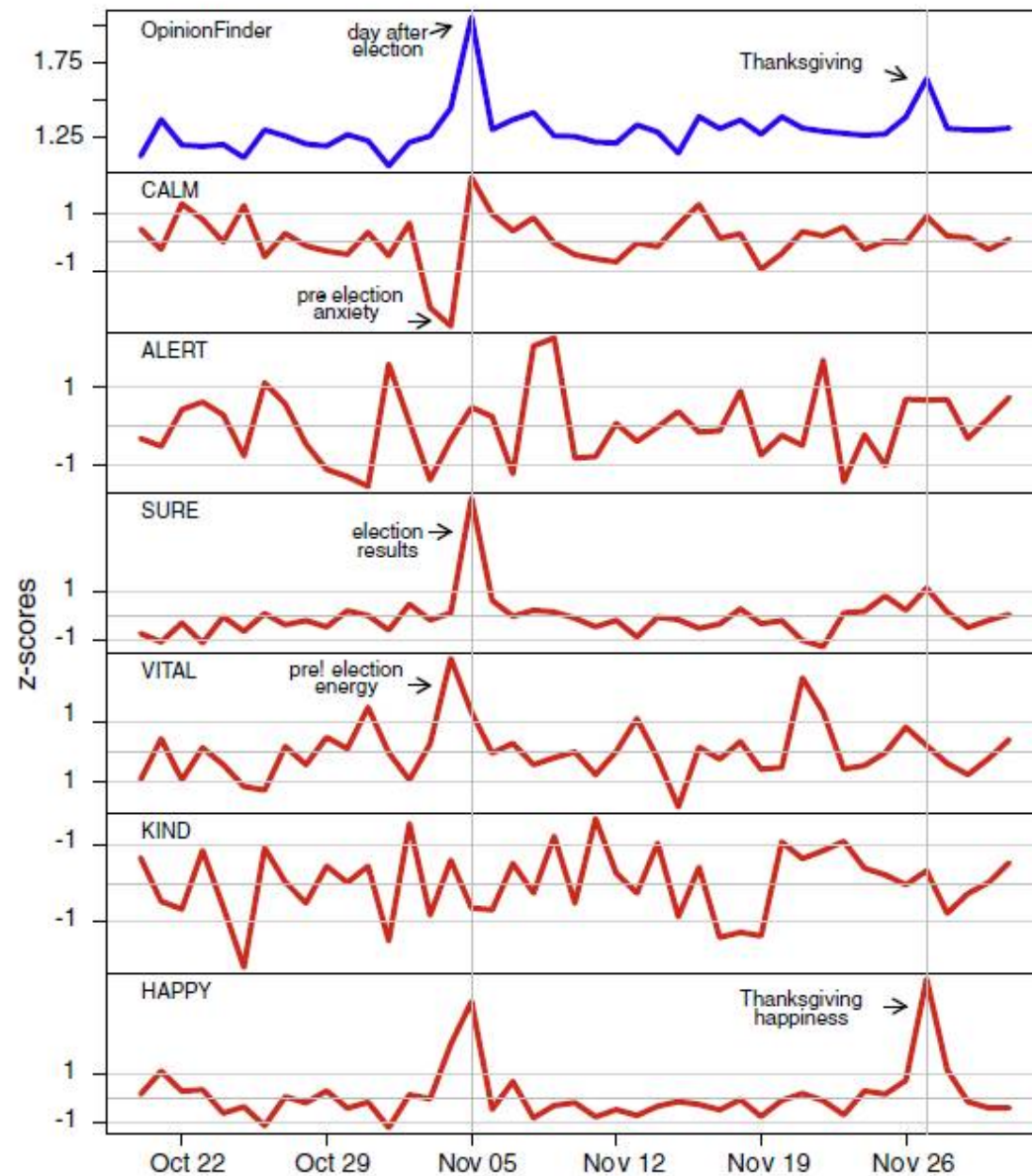


SOFNN(Self-Organizing Fuzzy Neural Network)  
자기 조직화 퍼지 신경망



# Analysis

- Tracking Mood states





# Analysis

**Table 1**

Multiple regression results for OpinionFinder vs. 6 GPOMS mood dimensions.

Parameters	Coeff.	Std. Err.	<i>t</i>	<i>P</i>
Calm ( $X_1$ )	1.731	1.348	1.284	0.205
Alert ( $X_2$ )	0.199	2.319	0.086	0.932
Sure ( $X_3$ )	3.897	0.613	6.356	<b>4.25e–08***</b>
Vital ( $X_4$ )	1.763	0.595	2.965	0.004*
Kind ( $X_5$ )	1.687	1.377	1.226	0.226
Happy ( $X_6$ )	2.770	0.578	4.790	<b>1.30e–05**</b>
Summary	Residual Std. Err.	Adj. $R^2$	$F_{6,55}$	<i>p</i>
	0.078	0.683	22.93	2.382e–13

\*  $p < 0.1$ .

\*\*  $p < 0.05$ .

\*\*\*  $p < 0.001$ .

# [잠깐] 논문을 이해하기 위한 사전지식

- P-value(유의확률)
  - $H_0$ (귀무가설)이 참이라는 가정하에 우연히, 실험 결과가 나왔을 확률
  - 예시: 신약이 기존 약보다 효과 있는지 실험한다고 가정
    - $H_0$ (귀무가설) – 신약과 기존약 별 차이 없다.
    - $H_1$ (대립가설) – 신약이 기존약보다 효과 있다.
    - 실험해보니 신약 복용한 환자가 빠르게 회복. 우연인가? 아니면 진짜 효과 있나?
  - P-value가 0.05보다 작다면(작으면 작을수록), 신약과 기존약 별 차이없다는 가정 하에 이러한 결과가 나올 확률이 매우 작다. 우연히 나온 것이 아니다.
  - 따라서, 귀무가설 기각. 신약의 효과가 있다.

# Analysis

**Table 2**

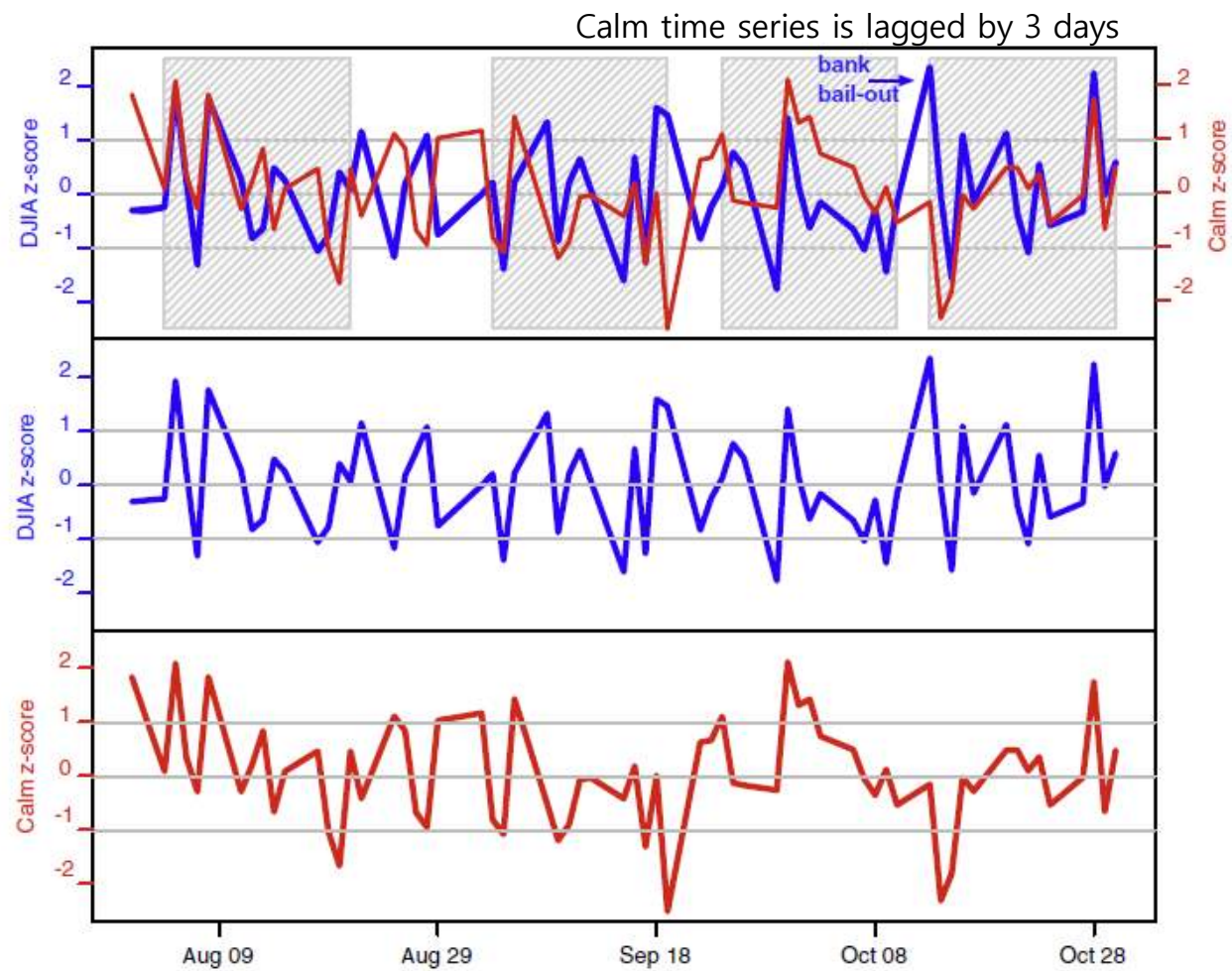
Statistical significance ( $p$ -values) of bivariate Granger-causality correlation between moods and DJIA in period February 28, 2008 to November 3, 2008.

Lag	OF	Calm	Alert	Sure	Vital	Kind	Happy
1 Day	<b>0.085*</b>	0.272	0.952	0.648	0.120	0.848	0.388
2 Days	0.268	<b>0.013**</b>	0.973	0.811	0.369	0.991	0.7061
3 Days	0.436	<b>0.022**</b>	0.981	0.349	0.418	0.991	0.723
4 Days	0.218	<b>0.030**</b>	0.998	0.415	0.475	0.989	0.750
5 Days	0.300	<b>0.036**</b>	0.989	0.544	0.553	0.996	0.173
6 Days	0.446	<b>0.065*</b>	0.996	0.691	0.682	0.994	<b>0.081*</b>
7 Days	0.620	0.157	0.999	0.381	0.713	0.999	0.150

\*  $p < 0.1$ .

\*\*  $p < 0.05$ .

# Analysis



# Discussion

- We investigated correlation between **public mood** measured from *twitter.com* and **DJIA**
  - **Changes in public mood state can be tracked from the content of Twitter feeds**
  - **Not in positive-negative mood But in “Calm” dim.**



*Think different*

# 감성분석 실습

이 한 준 교수

# 감성분석 실습

- ChatGPT를 활용해보자
  - 긍부정 분류에 좋은 데이터?

네, 한국어 영화 리뷰 데이터를 구할 수 있는 몇 가지 방법이 있습니다. 주로 사용되는 데이터셋은 네이버 영화 리뷰 데이터와 NSMC (Naver Sentiment Movie Corpus) 데이터셋입니다. NSMC 데이터셋은 이미 감정 라벨(긍정/부정)이 포함되어 있어 텍스트 분류 작업에 유용합니다.

## 1. NSMC (Naver Sentiment Movie Corpus)

NSMC는 네이버 영화 리뷰 데이터를 기반으로 감성 분석에 사용되는 데이터셋입니다. 이 데이터셋은 다음과 같은 링크에서 다운로드할 수 있습니다:

- 링크: [NSMC GitHub](#)
- 데이터 설명: NSMC는 긍정과 부정 리뷰를 포함한 텍스트와 해당 리뷰의 감정 라벨(0: 부정, 1: 긍정)을 제공합니다.

사용 예:

```
python

import pandas as pd

# 데이터 다운로드
url = "https://raw.githubusercontent.com/e9t/nsmc/master/ratings_train.txt"
train_data = pd.read_table(url)
train_data.head()
```

## 2. Korean Movie Review Dataset (K-Movie)

이 데이터셋은 다양한 영화 리뷰를 포함하고 있으며, 감성 분석, 영화 추천 시스템, 그리고 기타 NLP 작업에 활용될 수 있습니다. 다만, 이 데이터셋은 NSMC처럼 감정 라벨이 제공되지 않는 경우도 많아, 별도로 라벨링 작업이 필요할 수 있습니다.

- 링크: [K-Movie 리뷰 데이터셋](#)

## 3. Kakao Brain NLP Dataset

카카오는 다양한 NLP 관련 데이터셋을 제공하고 있습니다. 그중 일부는 영화 리뷰를 포함하고 있으며, 감성 분석 모델에 사용할 수 있습니다.

- 링크: [Kakao Brain NLP](#)

이 외에도 한국어 영화 리뷰 데이터는 공개된 여러 사이트에서 제공되며, 자체적으로 크롤링을 통해 수집할 수도 있습니다.



# 감성분석 실습

- 데이터 확인

```
import pandas as pd

# 데이터 다운로드
url = "https://raw.githubusercontent.com/e9t/nsmc/master/ratings_train.txt"
train_data = pd.read_table(url)
train_data.head()
```

	id	document	label
0	9976970	아 더빙.. 진짜 짜증나네요 목소리	0
1	3819312	흠...포스터보고 초딩영화줄....오버연기조차 가볍지 않구나	1
2	10265843	너무재밌었다그래서보는것을추천한다	0
3	9045019	교도소 이야기구먼 ..솔직히 재미는 없다..평점 조정	0
4	6483659	사이몬페그의 익살스런 연기가 돋보였던 영화!스파이더맨에서 늙어보이기만 했던 커스틴 ...	1

NSMC 데이터로 긍정 리뷰를 분류하는 모델을 만들고 싶어. 만들어줘

# 감성분석 실습

## • 분류 모델 구축

```
# 4. 모델 정의
model = Sequential([
    Embedding(input_dim=20000, output_dim=128, input_length=100),
    LSTM(64, return_sequences=True),
    LSTM(32),
    Dense(32, activation='relu'),
    Dropout(0.5),
    Dense(1, activation='sigmoid') # 이진 분류를 위한 시그모이드 활성화 함수
])

model.compile(loss='binary_crossentropy', optimizer=Adam(), metrics=['accuracy'])

# 5. 모델 학습
model.fit(X_train, y_train, epochs=3, batch_size=128, validation_split=0.2)

# 6. 평가
loss, accuracy = model.evaluate(X_train, y_train)
print(f"모델 정확도: {accuracy:.4f}")
```

NSMC 데이터셋을 사용하여 긍정과 부정 리뷰를 분류하는 모델을 만들기 위해, 먼저 데이터를 로드하고 전처리한 후, 텍스트 분류 모델을 구축하는 코드를 작성할 수 있습니다. 다음은 NSMC 데이터셋을 활용한 텍스트 분류 모델을 만드는 과정입니다.

### 1. 데이터 로드 및 전처리

우리는 NSMC 데이터셋을 로드하고, 텍스트 전처리 및 토큰화를 진행한 후 모델을 학습시킬 수 있습니다.

python

📄 복사 ✎ 편집

```
import pandas as pd
import numpy as np

from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout
from tensorflow.keras.optimizers import Adam

# 1. 데이터 로드
url_train = "https://raw.githubusercontent.com/e9t/nsmc/master/ratings_train.txt"
train_data = pd.read_table(url_train)

# 2. 데이터 전처리
train_data = train_data.dropna() # 결측치 제거
X_train = train_data['document'].values # 리뷰 텍스트
y_train = train_data['label'].values # 감정 라벨 (0: 부정, 1: 긍정)

# 3. 텍스트 전처리 및 토큰화
tokenizer = Tokenizer(num_words=20000)
tokenizer.fit_on_texts(X_train)

X_train = tokenizer.texts_to_sequences(X_train)
X_train = pad_sequences(X_train, maxlen=100) # 패딩 처리
```

# 감성분석 실습

- 분류 모델 구축
  - 시그모이드 값을 보여주도록 요구
    - 0.5를 기준으로 이하는 부정, 이상은 긍정
    - 실제 얼마나 강하게 예측하는지 확인

위 모델을 가지고 이제 새로운 리뷰를 키보드로 입력하면 긍부정을 분류할 수 있도록 코드 만들어줘

새로운 리뷰를 키보드로 입력받아 긍정 또는 부정으로 분류할 수 있도록 모델을 수정해 보겠습니다. 입력된 리뷰에 대해서는 전처리와 토큰화를 진행한 후, 학습된 모델을 사용하여 예측을 수행합니다.

아래는 그 구현 코드입니다:

```
# 7. 새로운 리뷰 예측
def predict_sentiment(review):
    # 리뷰 전처리
    sequence = tokenizer.texts_to_sequences([review])
    padded_sequence = pad_sequences(sequence, maxlen=100)

    # 예측
    prediction = model.predict(padded_sequence)

    # 결과 출력
    sentiment = "긍정" if prediction >= 0.5 else "부정"
    return sentiment, prediction[0][0]

# 8. 사용자가 키보드로 리뷰 입력받기
new_review = input("영화 리뷰를 입력하세요: ")

# 예측 결과 출력
sentiment, sigmoid_value = predict_sentiment(new_review)
print(f"리뷰: '{new_review}'\n예측된 감정: {sentiment} (시그모이드 값: {sigmoid_value:.4f})")
```

# 감성분석 실습

## • 성능평가와 테스트

- ChatGPT에서 코드 복사시 키보드 입력부분은 분리하여 붙여 넣을 것

```
import pandas as pd
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout
from tensorflow.keras.optimizers import Adam

# 1. 데이터 로드
url_train = "https://raw.githubusercontent.com/e9t/nsmc/master/ratings_train.txt"
train_data = pd.read_table(url_train)

# 2. 데이터 전처리
train_data = train_data.dropna() # 결측치 제거
X_train = train_data['document'].values # 리뷰 텍스트
y_train = train_data['label'].values # 감정 라벨 (0: 부정, 1: 긍정)

# 3. 토큰화 및 패딩
tokenizer = Tokenizer(num_words=20000)
tokenizer.fit_on_texts(X_train)

X_train = tokenizer.texts_to_sequences(X_train)
X_train = pad_sequences(X_train, maxlen=100) # 패딩 처리

# 4. 모델 정의
model = Sequential([
    Embedding(input_dim=20000, output_dim=128, input_length=100),
    LSTM(64, return_sequences=True),
    LSTM(32),
    Dense(32, activation='relu'),
    Dropout(0.5),
    Dense(1, activation='sigmoid') # 이진 분류를 위한 시그모이드 활성화 함수
])

model.compile(loss='binary_crossentropy', optimizer=Adam(), metrics=['accuracy'])

# 5. 모델 학습
model.fit(X_train, y_train, epochs=3, batch_size=128, validation_split=0.2)

# 6. 모델 평가
loss, accuracy = model.evaluate(X_train, y_train)
print(f"모형 정확도: (accuracy: {accuracy:.4f})")

# 7. 새로운 리뷰 예측 (함수)
def predict_sentiment(review):
    # 리뷰 전처리
    sequence = tokenizer.texts_to_sequences([review])
    padded_sequence = pad_sequences(sequence, maxlen=100)

    # 예측
    prediction = model.predict(padded_sequence)

    # 결과 출력
    sentiment = "긍정" if prediction[0] > 0.5 else "부정"
    return sentiment, prediction[0][0]

Epoch 1/3
938/938 [=====] - 193s 203ms/step - loss: 0.4526 - accuracy: 0.7744 - val_loss: 0.3820 - val_accuracy: 0.8144
Epoch 2/3
938/938 [=====] - 184s 196ms/step - loss: 0.3371 - accuracy: 0.8427 - val_loss: 0.3893 - val_accuracy: 0.8149
Epoch 3/3
938/938 [=====] - 187s 200ms/step - loss: 0.2907 - accuracy: 0.8664 - val_loss: 0.4336 - val_accuracy: 0.8119
4688/4688 [=====] - 97s 21ms/step - loss: 0.2622 - accuracy: 0.8789
모형 정확도: 0.8789
```

```
# 8. 사용자가 키보드로 리뷰 입력받기
new_review = input("영화 리뷰를 입력하세요: ")

# 예측 결과 출력
sentiment, sigmoid_value = predict_sentiment(new_review)
print(f"리뷰: '{new_review}'\n예측된 감정: {sentiment} (시그모이드 값: {sigmoid_value:.4f})")
```

영화 리뷰를 입력하세요:

# 감성분석 실습

- 성능평가와 테스트
  - 가상의 리뷰를 만들어 넣어보자

```
# 8. 사용자가 키보드로 리뷰 입력받기
new_review = input("영화 리뷰를 입력하세요: ")

# 예측 결과 출력
sentiment, sigmoid_value = predict_sentiment(new_review)
print(f"리뷰: '{new_review}'\n예측된 감정: {sentiment} (시그모이드 값: {sigmoid_value:.4f})")
```

```
영화 리뷰를 입력하세요: 음악이며 연기며 하나부터 열까지 찐다
1/1 [=====] - 1s 662ms/step
리뷰: '음악이며 연기며 하나부터 열까지 찐다'
예측된 감정: 긍정 (시그모이드 값: 0.6312)
```

# 감성분석 실습

## • 모델 변경

- LSTM 외에 다른 모델을 사용하는 코드를 ChatGPT에게 부탁하자
- 모델 제작에 코드 작성 실력은 이제 걱정할 필요가 없어짐

LSTM 외에도 최근 텍스트 분류에서 좋은 성능을 보이는 모델로 **BERT**(Bidirectional Encoder Representations from Transformers) 모델을 사용할 수 있습니다. BERT는 문맥을 잘 이해하는 모델로, 텍스트 분류 문제에서 매우 효과적입니다.

이 코드를 사용하여 BERT 모델을 기반으로 텍스트 분류를 수행할 수 있도록 바꿔보겠습니다. BERT를 사용하려면 `transformers` 라이브러리와 `tensorflow`의 Keras API를 함께 활용해야 합니다.

다음은 BERT를 사용한 긍정/부정 리뷰 분류 모델 코드입니다.

### 1. BERT를 활용한 긍정/부정 분류 모델 코드

```
python

import pandas as pd
import numpy as np
from transformers import BertTokenizer, TFBertForSequenceClassification
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping

# 1. 데이터 로드
url_train = "https://raw.githubusercontent.com/e9t/nsmc/master/ratings_train.txt"
train_data = pd.read_table(url_train)

# 2. 데이터 전처리
train_data = train_data.dropna() # 결측치 제거
X_train = train_data['document'].values # 리뷰 텍스트
y_train = train_data['label'].values # 감정 라벨 (0: 부정, 1: 긍정)

# 3. BERT 토큰라이저 준비
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
```

팀 플

이 한 준 교수

# 이번 학기 목표 포트폴리오

- 한국디지털콘텐츠학회
  - 2025년 하계종합학술대회 or 대학생논문경진대회
  - <https://dcs.or.kr/homepage/custom/submit>

\* 개최기간 : 2025년 7월 3일(목) ~ 7월 5일(토)

\* 장 소 : 오션스위츠제주호텔

## 논문분야

1. 일반(전체) : 디지털 전환, AI와 콘텐츠 융합 기술  
세부 분과 1 : 메타버스 및 융합형 플랫폼  
세부 분과 2 : 인공지능 및 Data Science  
세부 분과 3 : 디지털 트윈 및 XR(VR/AR/MR) 콘텐츠  
세부 분과 4 : ICT 융·복합 기술 전분야  
세부 분과 5 : 산업융합분야 (기타)
2. 대학생 : ICT 관련 전 분야 및 캡스톤디자인 작품  
※ 위와 같은 세부분과로 모집하지만 분야에 제한을 두지 않음.

## 논문투고

1. 일반 : 구두, 포스터 구분, 우수논문 시상
2. 대학생 : 포스터, 대상/금상/은상/동상 상장 수여
3. 논문제출 기간 : 3월 24일(월) ~ 6월 8일(일) 23시 59분까지
4. 논문양식 : 일반, 대학생 구분 / 2단 편집으로 1~2페이지 제출  
※ 학회 홈페이지>자료실>주요양식 [46번] 게시글에서 다운로드 가능
5. 논문투고 : 학회 홈페이지 > 국내학술행사 > 참가신청 > 논문투고  
※이메일로는 논문 제출 및 접수받지 않음
6. 채택통보 : 2025년 6월 2일(월)부터 순차 발송

## 논문발표

1. 논문 발표일 : 7월 3일(목)-5일(토) 오전/오후 (발표 시간은 추후 공지 예정)
2. 구두 발표 : 10분 발표
3. 포스터 발표 : PPT 5분 발표 및 질의응답
4. 온라인발표 : 학술대회 당일 Zoom으로 발표 및 질의응답



# 이번 학기 목표 포트폴리오

- 한국디지털콘텐츠학회

- 2025년 하계종합학술대회 or 대학생논문경진대회
- <https://dcs.or.kr/homepage/custom/submit>

등 록 비	<p><b>사전등록 기간 : ~ 6월 15일(일)까지</b></p> <p>1. 일반 : 회원 10만원, 비회원 12만원</p> <p>2. 대학생 : 6만원</p> <p>3. 논오서(비저자 및 단순참가) : 6만원</p> <p>※ 홈페이지 가입회원은 정회원이 아니므로 홈페이지에서 회원가입하고 정회원비를 납부해야 함</p> <p>※ 현장등록 불가능 (단, 과제 카드의 경우는 학회로 문의)</p> <p>※ 대학생논문경진대회 논문: 온라인 "발표"를 원칙으로 하되, 오프라인 "참석"도 가능함.</p>
등록방법	<p><b>6월 15일(일)까지 사전등록 및 등록비납부</b></p> <p>(우리은행, 1005-504-315334 (사)한국디지털콘텐츠학회)</p> <p>(전자계산서 발행 원하는 경우 사전등록 페이지에서 신청)</p> <p>※사전등록 : 학회 홈페이지 &gt; 국내학술행사 &gt; 참가신청 &gt; 사전등록(등록비 납부)</p>
장 소	오션스위츠제주호텔, 제주특별자치도, 064-720-6000

# 이번 학기 목표 포트폴리오

- 최종적으로는 KCI(Korea Citation Index) 등재 학술지 게재 목표
- 학기말 이후 하계 방학기간 중 마무리 하여 투고
  - 빠르게 진행된다면 1학기 중 마무리 투고

# 다음주부터는 팀별 브리핑

- 자유로운 형식으로 진행 과정 리포트
- 진행 과정에서 어려움과 조언 필요한 부분 공유
- 서로 벤치마킹 및 리뷰 기회
- 어떤 관련 논문을 참고했는지 이야기할 것
  - 다양한 논문을 볼 것