# 딥러닝을 이용한 자연어 처리

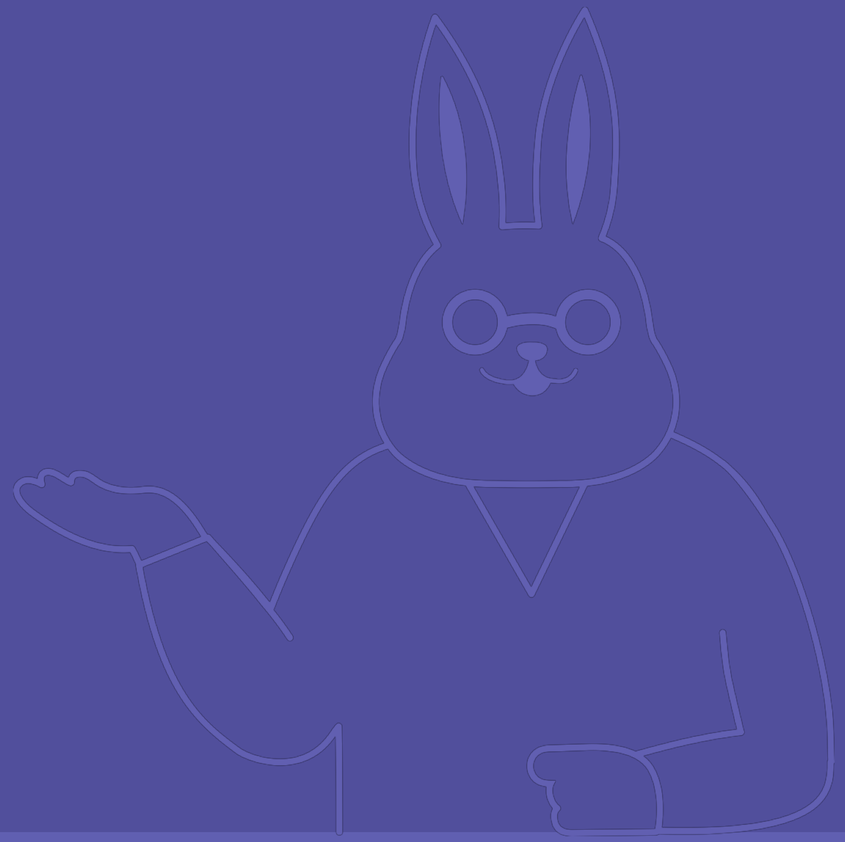
## 02 감정 분석 서비스

Confidential all rights reserved

# 목차

01

# 감정 분석 서비스



Confidential all rights reserved

## 텍스트 데이터의 종류


뉴스, 백과 사전 같은 텍스트는 **객관적인 정보**를 제공

## 텍스트 데이터의 종류

| 문장 | 감정 |
| --- | --- |
| 영상미가 뛰어나고 너무너무 재미있었어요… | 기쁨 |
| 배우들의 뛰어난 연기… | 기쁨 |
| 허무한 결말… | 분노 |

리뷰, 소설 같은 텍스트는 **저자의 주관적인 평가나 감정**을 표현

## 감정 분석이란

| 문장 | 감정 |
| --- | --- |
| 영상미가 뛰어나고 너무너무 재미있었어요… | 기쁨 |
| 배우들의 뛰어난 연기… | 기쁨 |
| 허무한 결말… | 분노 |

대량의 텍스트가 있는 경우, **일일이** 데이터를 하나씩 살펴보고 판단하기 어려움

## 감정 분석이란

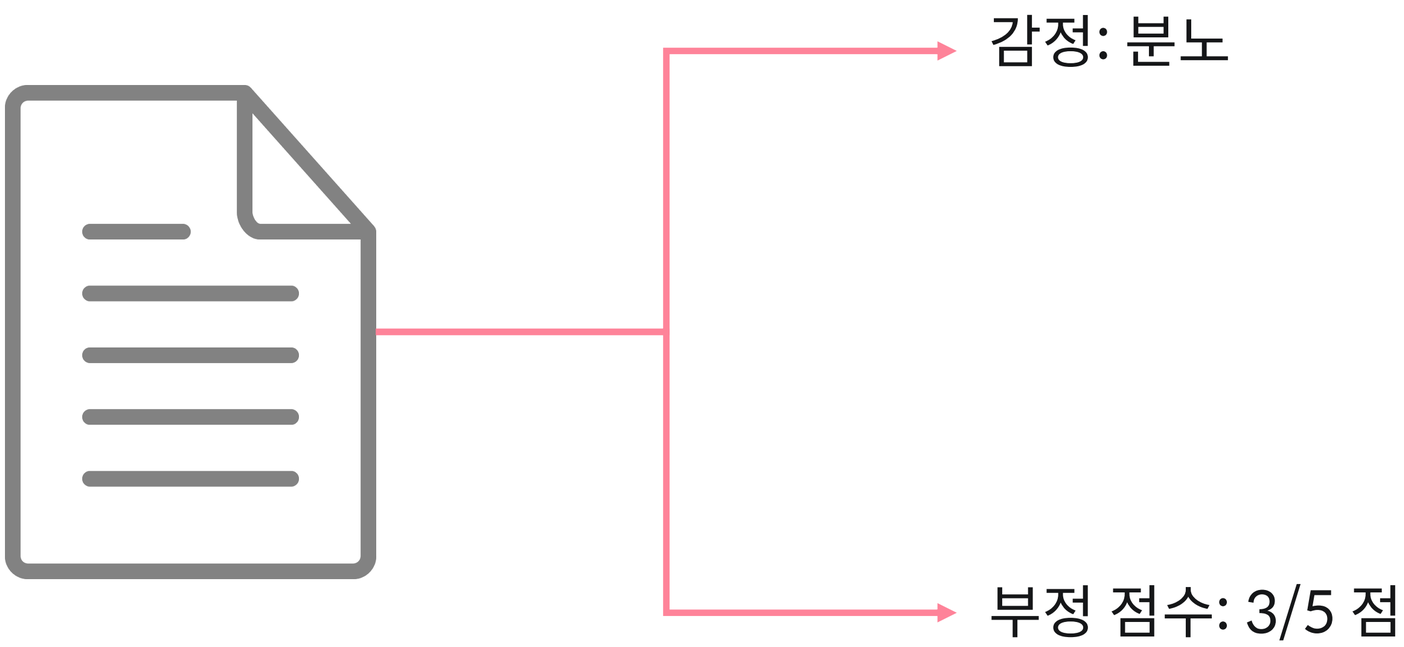
| 문장 | 감정 |
| --- | --- |
| 영상미가 뛰어나고 너무너무 재미있었어요… | 기쁨 |
| 배우들의 뛰어난 연기… | 기쁨 |
| 너무 흥미진진한 구성… | 기쁨 |

비슷한 감정을 표현하는 문서는 **유사한 단어 구성 및 언어적 특징**을 보일 것을 가정

## 감정 분석이란

**감정 분석(Sentiment analysis)**은 텍스트 내에 표현되는
감정 및 평가를 식별하는 자연어 처리의 한 분야

## 감정 분석이란




텍스트 내 **감정을 분류**하거나 **긍정/부정의 정도**를 점수화

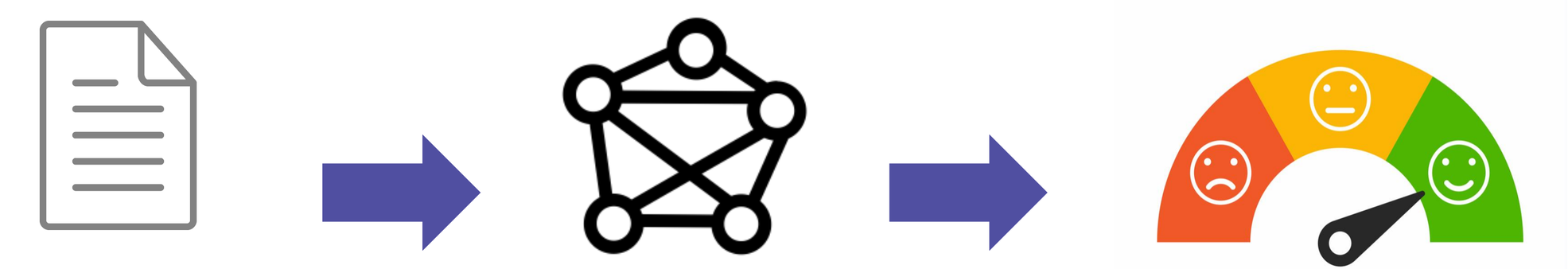
## 감정 분석 서비스

| 문장 | 감정 |
| --- | --- |
| 영상미가 뛰어나고 너무너무 재미있었어요… | 기쁨 |
| 허무한 결말… | 분노 |
| 배우들의 뛰어난 연기… | 기쁨 |

.
.
.

머신러닝 기반 감정 분석 서비스의 경우, 데이터를 통한 **모델 학습**부터 시작

## 감정 분석 서비스

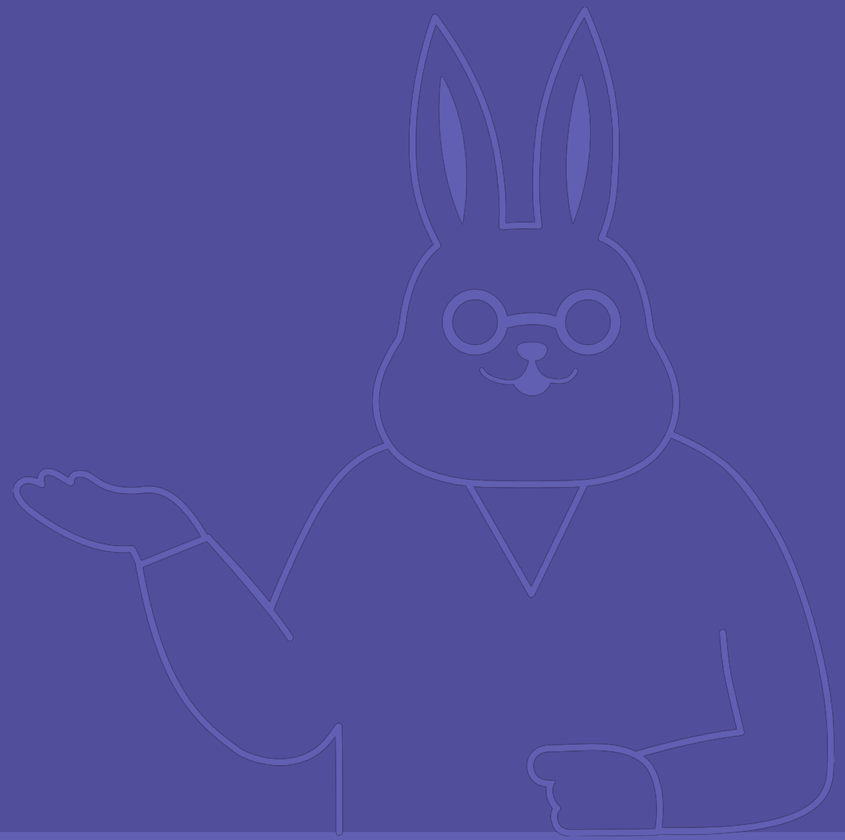


신규 텍스트 학습된 모델 감정 예측

학습된 머신러닝 모델을 통해 **신규 텍스트의 감정**을 예측

02

# 나이브 베이즈



Confidential all rights reserved

## 나이브 베이즈의 원리

**[텍스트 1]** : 영상미가 | 뛰어나고 | 너무너무 | 재미있었어요

$$P(\text{감정} \mid \text{텍스트}) = ?$$

**나이브 베이즈 기반 감정 분석**은 주어진 텍스트가
특정 감정을 나타낼 확률을 예측하는 문제로 정의

## 나이브 베이즈의 원리

$$P(\text{감정} \mid \text{텍스트}) = \frac{P(\text{텍스트} \mid \text{감정}) \times P(\text{감정})}{P(\text{텍스트})}$$

**베이즈 정리**를 사용하여 텍스트의 감정 발생 확률을 추정

## 나이브 베이즈의 원리

**[텍스트 1]** : 영상미가 | 뛰어나고 | 너무너무 | 재미있었어요

**[텍스트 1의 감정]** : 해당 감정 내 단어들이 발생할 가능성 × 감정의 발생 확률

**감정의 발생 확률**과 텍스트를 구성하는
**단어들의 가능도(likelihood)**로 텍스트의 감정을 예측

## 단어의 가능도

$$\hat{P}(\text{단어} \mid \text{감정}) = \frac{(\text{감정 내 단어의 빈도수})}{(\text{감정 내 모든 단어의 빈도수})}$$

$$\hat{P}(\text{"재미있었어요"} \mid \text{기쁨}) = \frac{(\text{기쁨을 표현하는 문서 내 "재미있었어요"의 빈도수})}{(\text{기쁨을 표현하는 문서 내 모든 단어의 빈도수})}$$

텍스트 데이터에서는 가능도는 단어의 **빈도수**로 추정

## 감정의 발생 확률

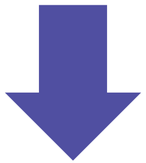
$$P(\text{감정}) = \frac{(\text{해당 감정을 표현하는 문서의 수})}{(\text{데이터 내 모든 문서의 수})}$$

$$P(\text{기쁨}) = \frac{(\text{기쁨을 표현하는 리뷰의 수})}{(\text{전체 리뷰의 수})}$$

감정의 발생 확률은 주어진 텍스트 데이터 내 해당 감정을 표현하는 **문서의 비율**로 추정

## 텍스트의 감정

[텍스트 1] : 영상미가 | 뛰어나고 | 너무너무 | 재미있었어요



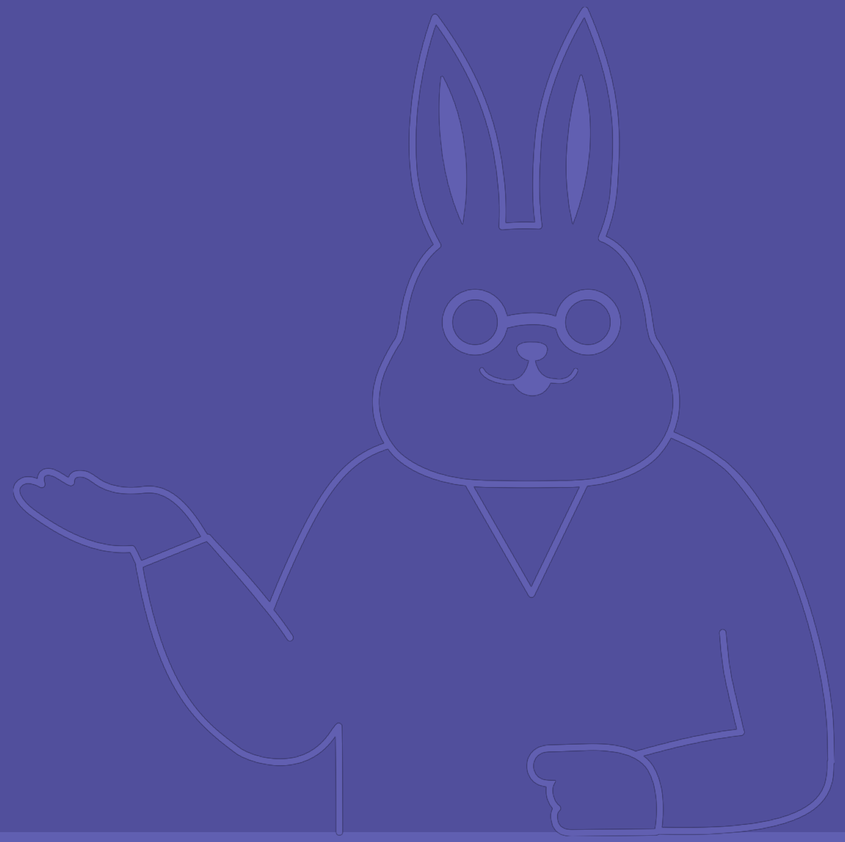
[텍스트 1이 **기쁨**을 나타낼 확률] : $\hat{P}$("영상미가"| 기쁨) × … × $\hat{P}$("재미있었어요"| 기쁨) × P(기쁨)

[텍스트 1이 **분노**를 나타낼 확률] : $\hat{P}$("영상미가"| 분노) × … × $\hat{P}$("재미있었어요"| 분노) × P(분노)

텍스트의 감정별 확률값 중 **최대 확률값**을 나타내는 감정을 해당 문서의 감정으로 예측

03

# 나이브 베이즈 기반 감정 예측



Confidential all rights reserved

## 스무딩 (smoothing)

학습 데이터 내 재미있었어요의 빈도 = 0

$$\hat{P}(\text{"재미있었어요"} \mid \text{기쁨}) = \frac{(\text{기쁨을 표현하는 문서 내 "재미있었어요"의 빈도수})}{(\text{기쁨을 표현하는 문서 내 모든 단어의 빈도수})} = 0$$

학습 데이터 내 존재하지 않은 단어가 포함된 문장의 **감정 발생 확률은 0**

## 스무딩 (smoothing)

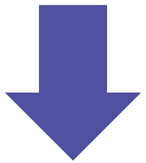
학습 데이터 내 재미있었어요의 빈도 = 0

$$\hat{P}(\text{"재미있었어요"} \mid \text{기쁨}) = \frac{(\text{기쁨을 표현하는 문서 내 재미있었어요의 빈도수}) + 1}{(\text{기쁨을 표현하는 문서 내 모든 단어의 빈도수}) + 1}$$

**스무딩(smoothing)을** 통해 학습 데이터 내 존재하지 않은 단어의 빈도수를 보정

## 소수

[텍스트 1] : 영상미가 | 뛰어나고 | 너무너무 | 재미있었어요



[텍스트 1이 **기쁨**을 나타낼 확률] : $0.52 \times \cdots \times 0.75 \times 0.22$

[텍스트 1이 **분노**를 나타낼 확률] : $0.1 \times \cdots \times 0.001 \times 0.35$

단어의 감정별 가능도와 감정의 발생 확률은 모두 **소수**로 표현

## 소수

$$0.1 \times 0.1 = 0.01$$

$$0.1 \times 0.1 \times 0.1 = 0.001$$

$$0.1 \times 0.1 \times 0.1 \times 0.1 = 0.0001$$

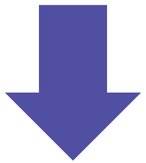
$$0.1 \times 0.1 \times 0.1 \times 0.1 \times 0.1 = 0.00001$$

$$\vdots$$

연속적으로 **소수**를 곱하면 결괏값은 끊임없이 감소

## 소수

[텍스트 1] : 동해물과 | 백두산이 | 마르고 | 닳도록 …



[텍스트 1이 **기쁨**을 나타낼 확률] : $0.52 \times \cdots \times 0.12 \times 0.12 = 0.00000000000000000000037291$

단어의 수가 많아질수록 텍스트의 확률값은 컴퓨터가 처리할 수 있는

**소수점의 범위보다 작아질 수 있음**

## 로그

$$\log_{10}(0.1 \times 0.1) = \log_{10}(0.1) + \log_{10}(0.1) = -2$$

$$\log_{10}(0.1 \times 0.1 \times 0.1) = \log_{10}(0.1) + \log_{10}(0.1) + \log_{10}(0.1) = -3$$

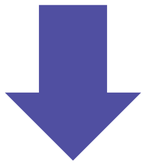
$$\log_{10}(0.1 \times 0.1 \times 0.1 \times 0.1) = \log_{10}(0.1) + \log_{10}(0.1) + \log_{10}(0.1) + \log_{10}(0.1) = -4$$

.
.
.

로그를 사용하면 끊임없이 숫자가 작아지는 것을 방지

## 최종 나이브 베이즈

[텍스트 1] : 영상미가 | 뛰어나고 | 너무너무 | 재미있었어요

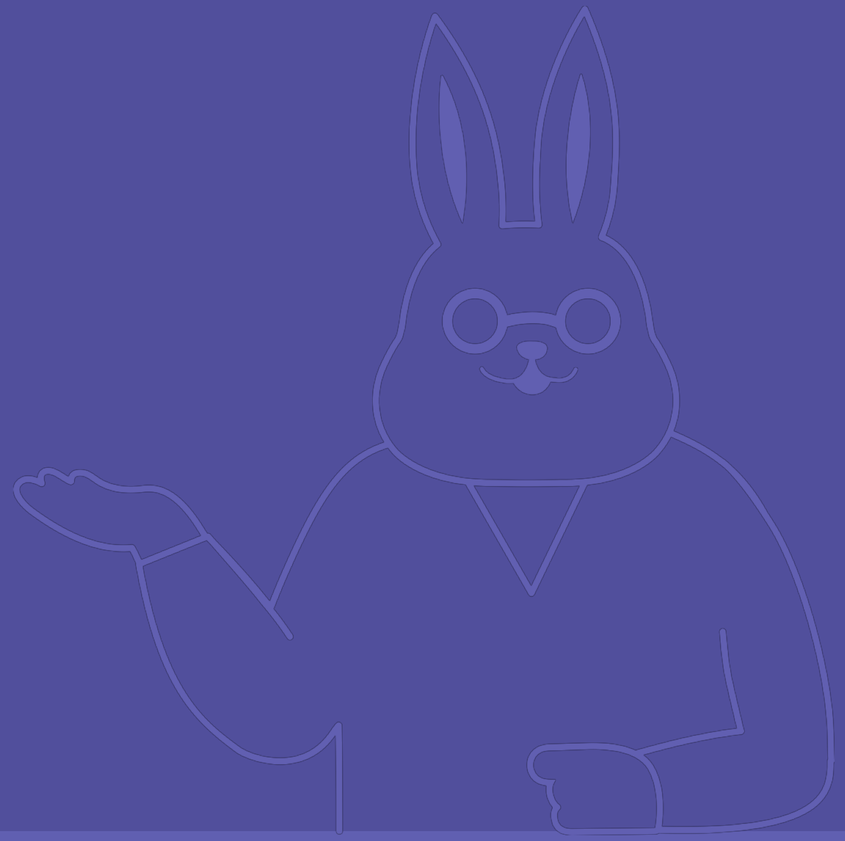


[텍스트 1이 **기쁨**을 나타낼 확률] : $\log(\hat{P}(\text{"영상미가"}|\text{기쁨})) + \cdots + \log(\hat{P}(\text{"재미있었어요"}|\text{기쁨})) + \log(P(\text{기쁨}))$

**로그 확률값의 합**으로 텍스트의 감정을 예측

04

# scikit-learn을 통한 나이브 베이즈 구현



Confidential all rights reserved

## scikit-learn


**scikit-learn**은 각종 데이터 전처리 및 머신 러닝 모델을
간편한 형태로 제공하는 파이썬 라이브러리

## scikit-learn

Example

```
from sklearn.feature_extraction.text import CountVectorizer

doc = ["i am very happy", "this product is really great"]
emotion = ["happy", "excited"]

cv = CountVectorizer()
csr_doc_matrix = cv.fit_transform(X_train)

#  각 단어 및 문장별 고유 ID 부여 및 단어의 빈도수를 계산
print(csr_doc_matrix) #  (0, 0)     1, (0, 7)       1
```

## scikit-learn

Example

```
from sklearn.naive_bayes import MultinomialNB

doc = ["i am very happy", "this product is really great"]
emotion = ["happy", "excited"]

clf = MultinomialNB()

# CountVectorizer로 변환된 텍스트 데이터를 사용
clf.fit(csr_doc_matrix, emotion)
```

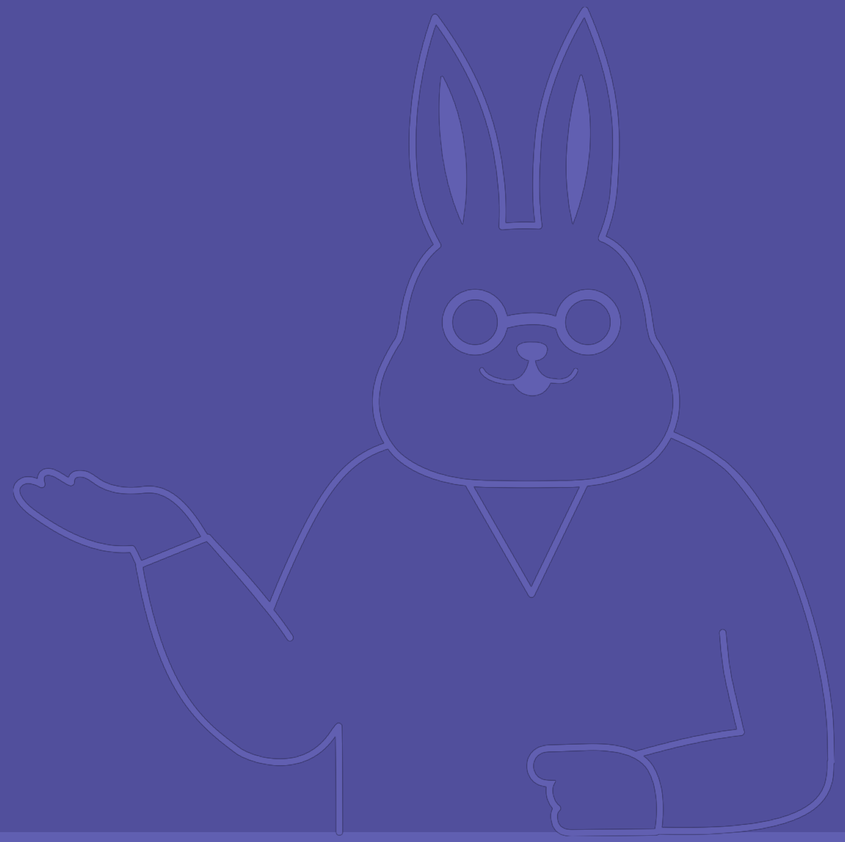
## scikit-learn

Example

```
from sklearn.naive_bayes import MultinomialNB

test_doc = ["i am really great"]

# 학습된 CountVectorizer 형태로 변환
transformed_test = cv.transform(test_doc)
pred = clf.predict(doc_vector)

print(pred) # array(['excited'], dtype='<U7')
```

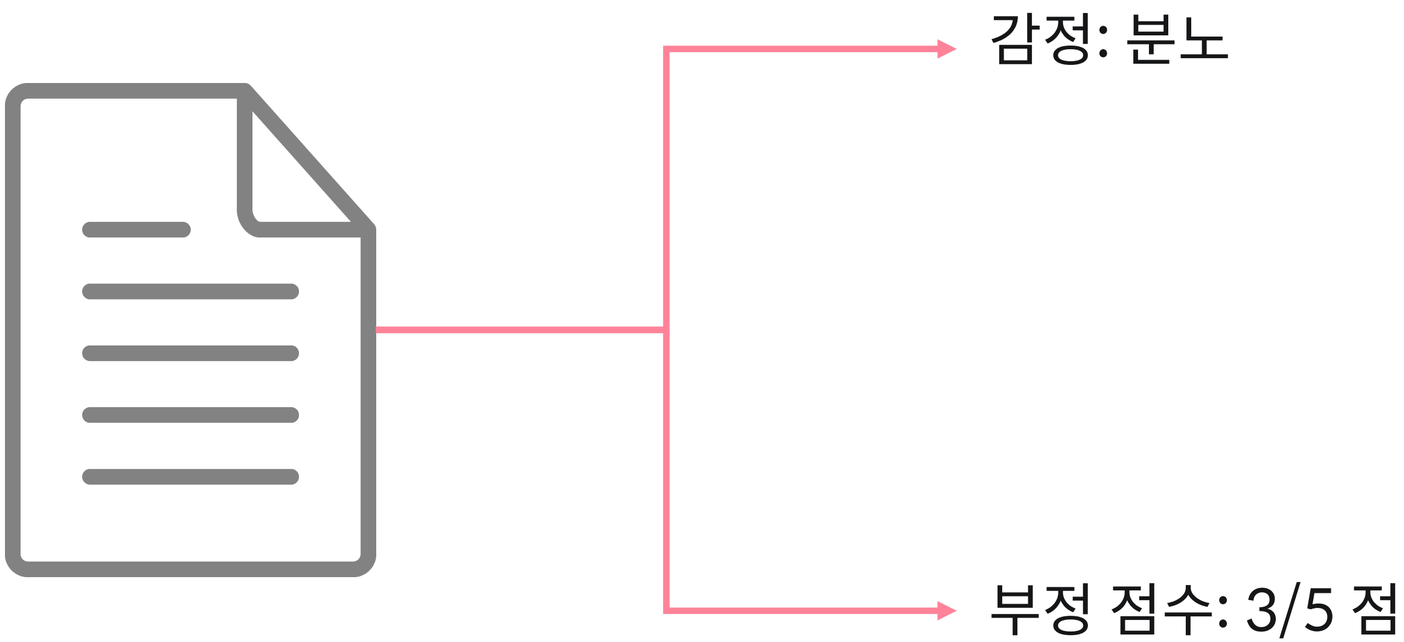
05

# 기타 감정 분석 방법



Confidential all rights reserved

## 감정 분석




감정 분석은 **지도 학습(supervised learning) 기반**의 분류 및 예측의 문제

## 감정 분석 + 머신러닝

| 문장 | 감정 |
| --- | --- |
| 영상미가 뛰어나고 너무너무 재미있었어요… | 기쁨 |
| 허무한 결말… | 분노 |
| 인생 최고의 영화… | 감동 |

학습 데이터에 감정만 존재하면 **머신러닝 알고리즘 학습**이 가능

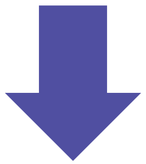
## 감정 분석 + 머신러닝

| | | |
|---|---|---|
| 엘리스 | | [0.23, 0.401, 0.482, 0.42, 0.99 … ] |
| 자연어 | | [0.03, 0.101, 2.682, 0.62, 0.08 … ] |
| 처리 | → | [1.03, 0.222, 0.382, 0.32, 1.28 … ] |
| . | | . |
| . | | . |
| . | | . |

**임베딩 벡터**를 사용하여, 머신러닝 알고리즘 적용이 가능

## 예시: 평균 임베딩 벡터

[텍스트 1] : 영상미가 | 뛰어나고 | 너무너무 | 재미있었어요 | …



영상미 : [0.12, 0.24, 0.913 …]

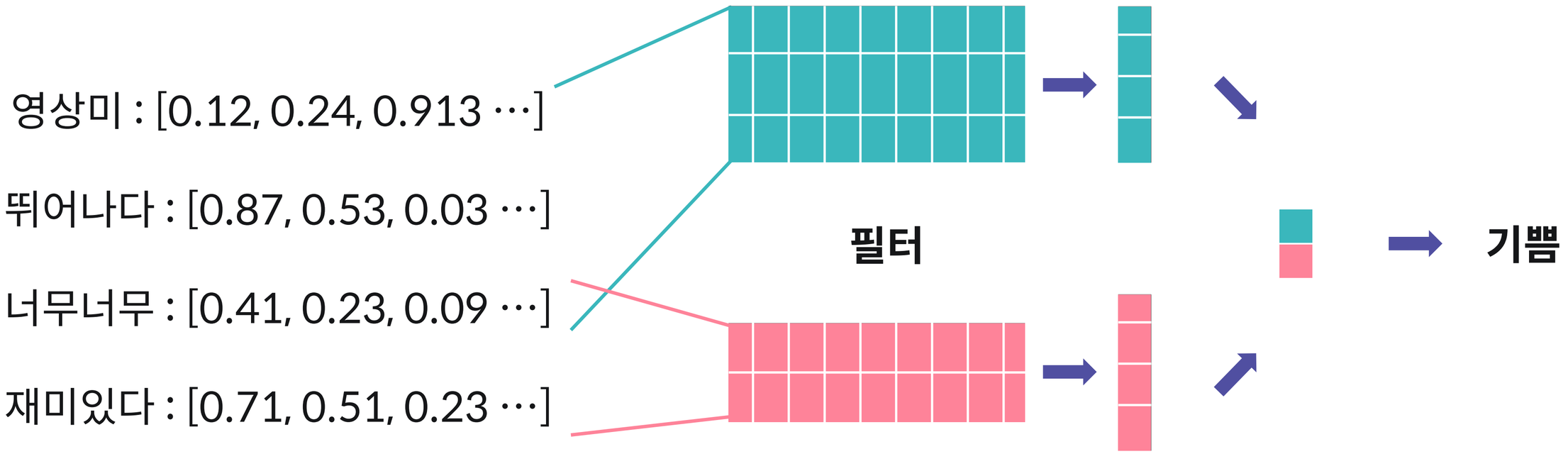
+

뛰어나다 : [0.87, 0.53, 0.03 …]

+

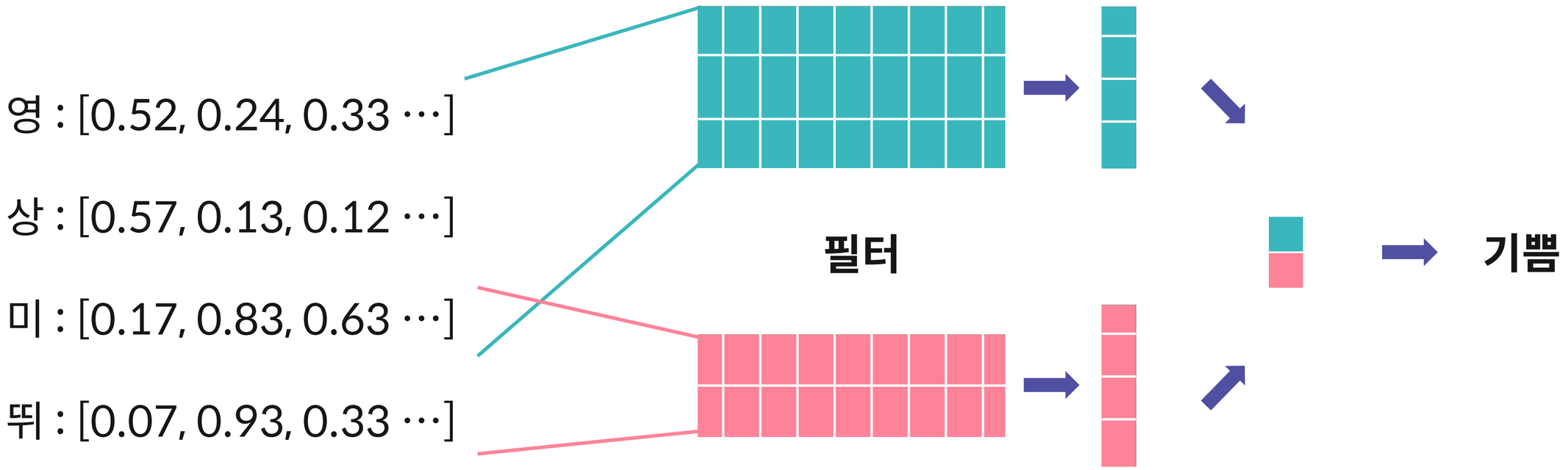
가장 간단한 방법으로 **단어 임베딩 벡터의 평균**을 사용

## 예시: CNN



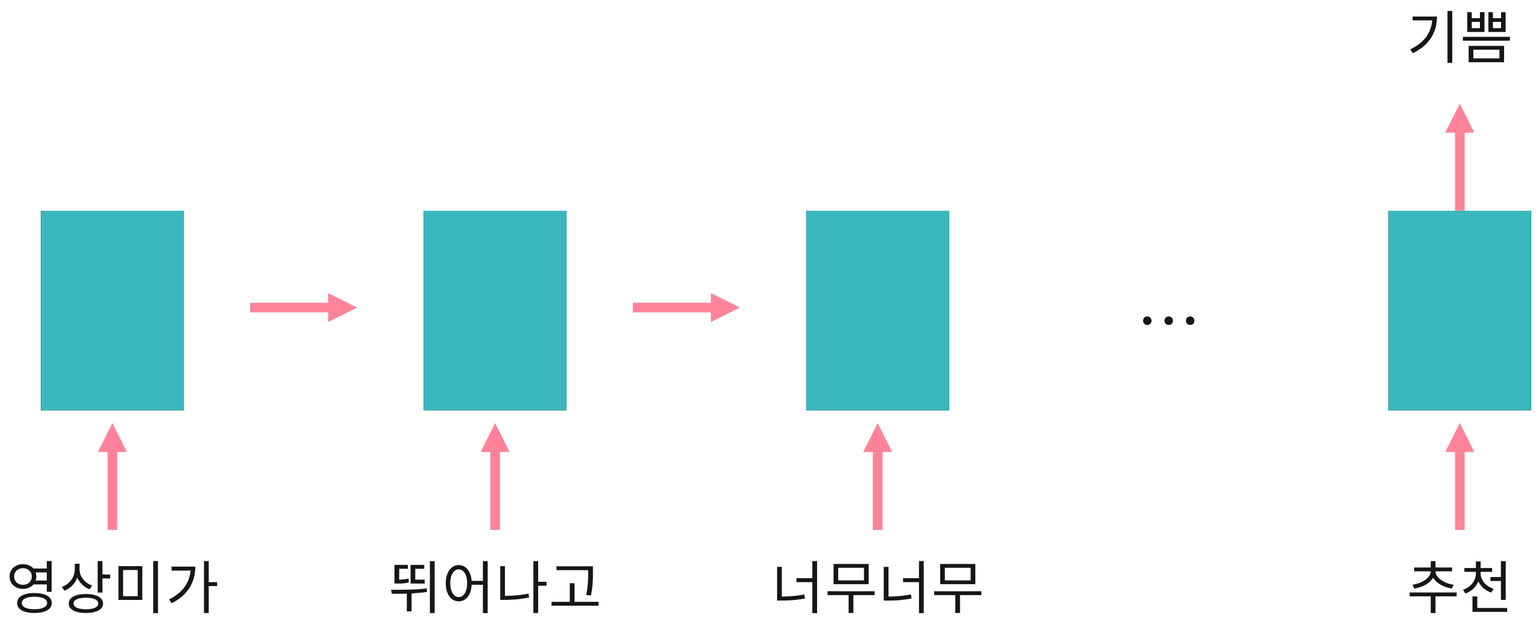

**단어 임베딩 벡터에 필터**를 적용하여 CNN 기반으로 감정 분류

## 예시: CNN



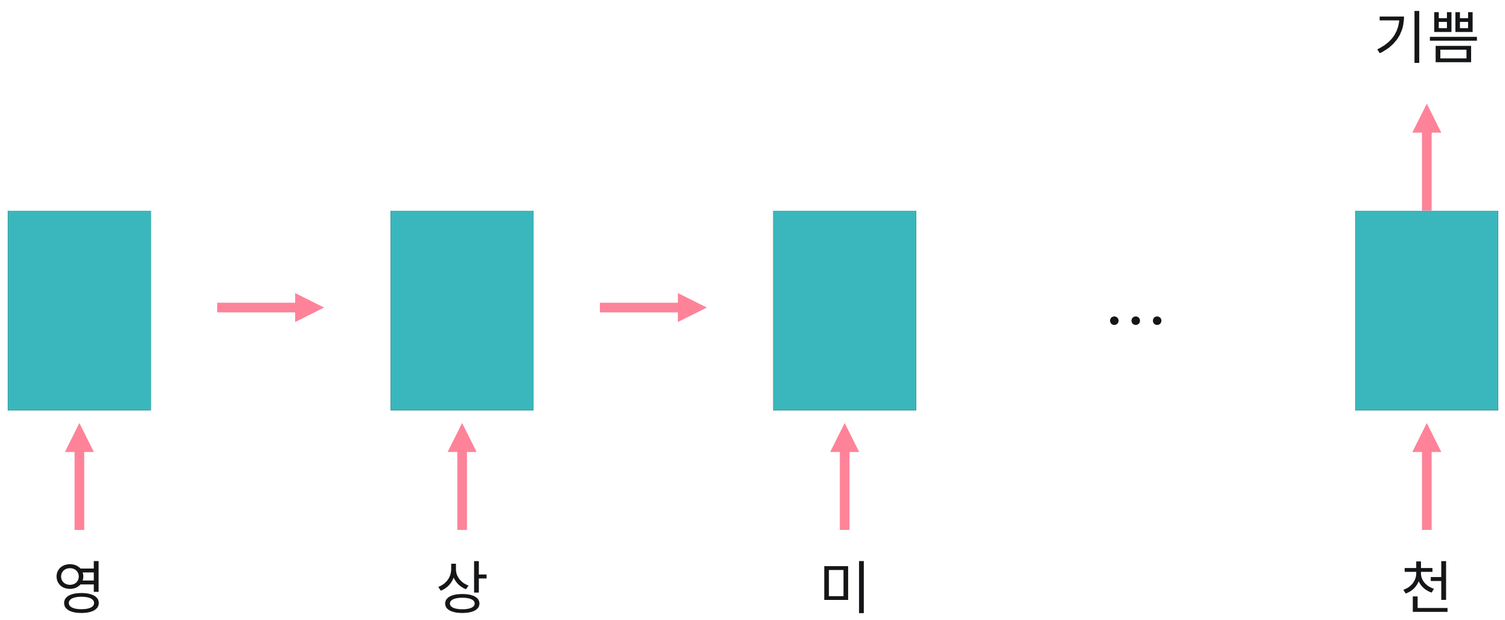

**문자 임베딩 벡터에 필터**를 적용하여 CNN 기반으로 감정 분류

## 예시: RNN




**LSTM, GRU**를 활용하여 RNN 기반으로 분류 및 예측

## 예시: RNN




**문자 단위로 단어를 분리**하여 RNN 기반으로 분류 및 예측