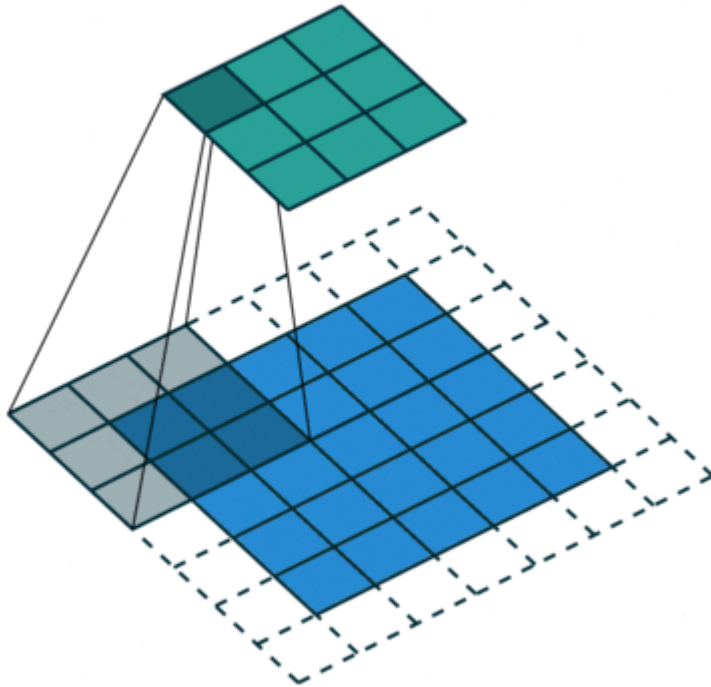


# **M6. CNN for NLU**

## [Recall] Convolution 연산



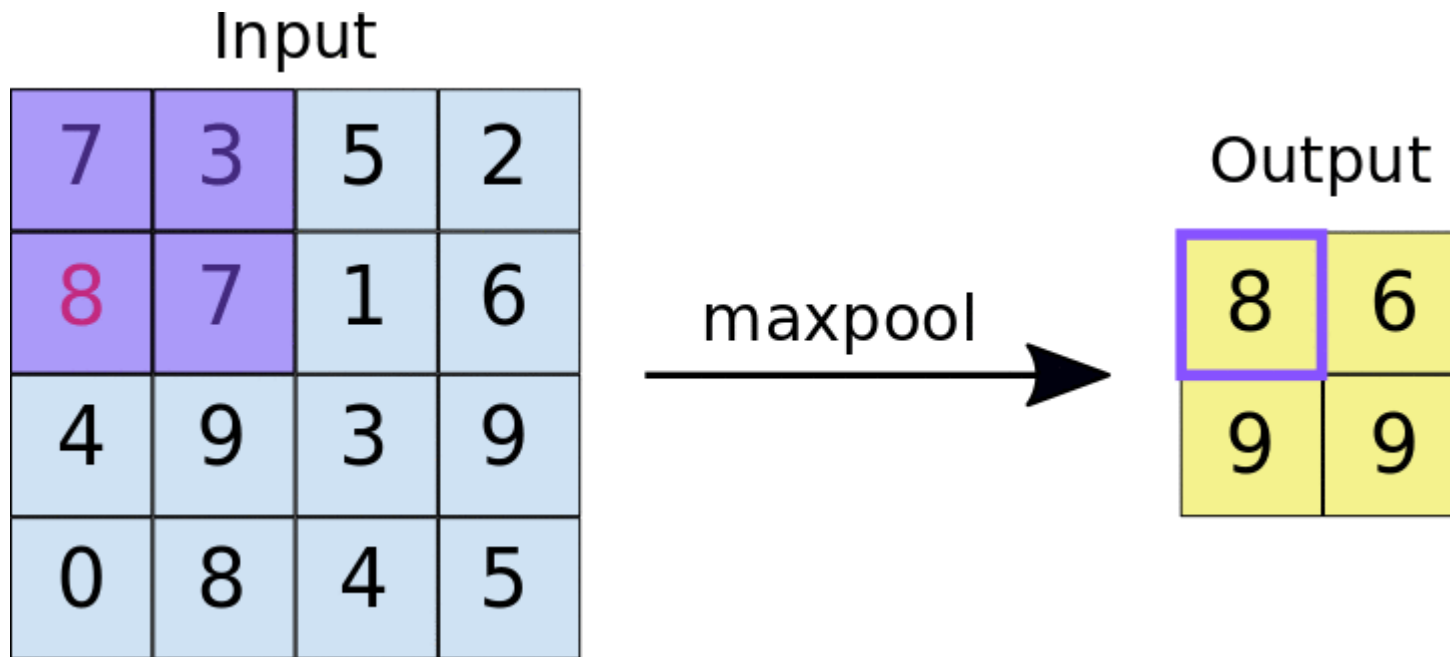
1 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	0	0
0 <sub>x0</sub>	1 <sub>x1</sub>	1 <sub>x0</sub>	1	0
0 <sub>x1</sub>	0 <sub>x0</sub>	1 <sub>x1</sub>	1	1
0	0	1	1	0
0	1	1	0	0

4		

Convolved  
Feature

- $n \times m$  크기의 필터가 데이터를 돌며 지역적인 정보를 가공해 feature 추출

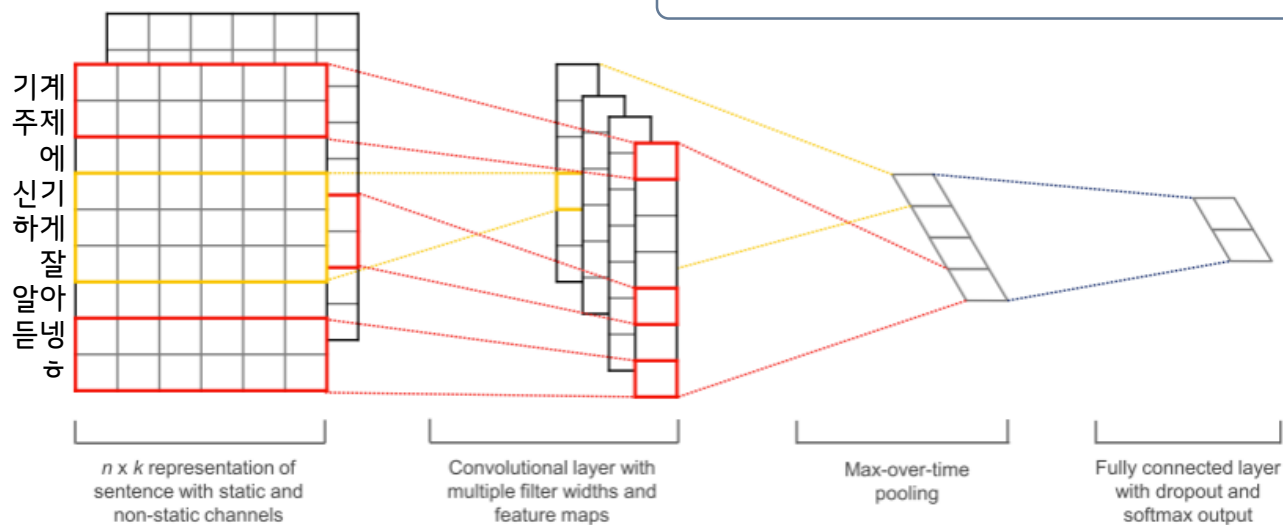
[Recall] Max pooling



- $a \times b$  지역 내에서 최대값만을 가지고 오는 연산

2 dimensional convolution layer을 사용해 문장 representation 을 생성하는 방법

- 이미지를 처리하듯 텍스트를 처리하는 방식
- 위치 무관하게 문장의 특징을 뽑아내어 분류
- 병렬처리가 가능해 속도가 빠름



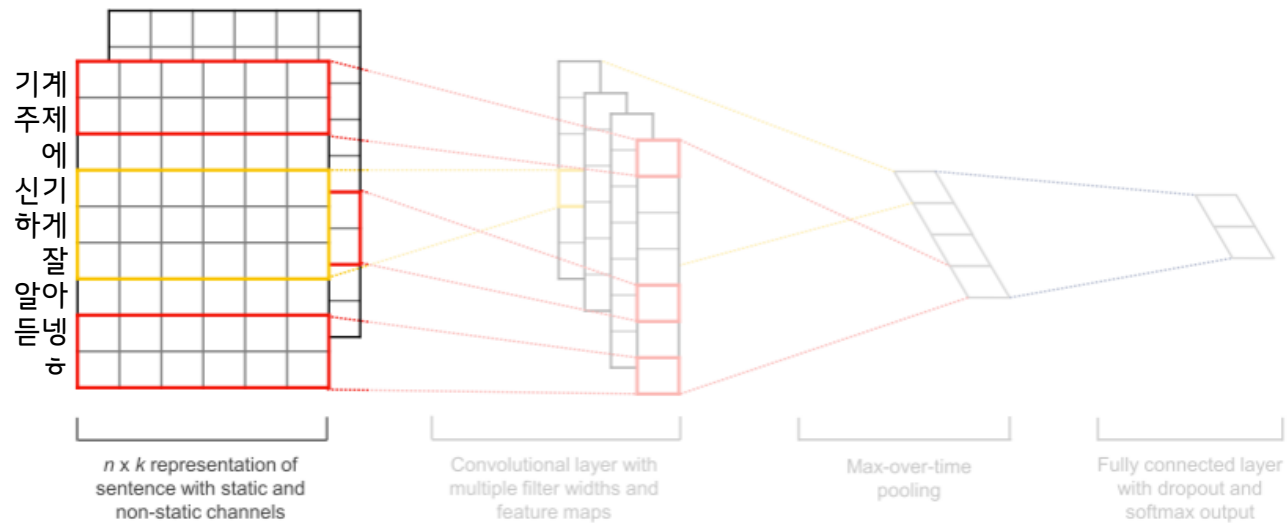
Convolutional Neural Networks for Sentence Classification (Yoon Kim, 2014)

\*\* 위의 예시는 token에 대해 convolution을 적용했지만, character 단위로 적용하는 character CNN도 가능함.

## CNN Step by Step

## 1) INPUT

: 시퀀스 길이  $\times$  단어 임베딩  
차원의 매트릭스



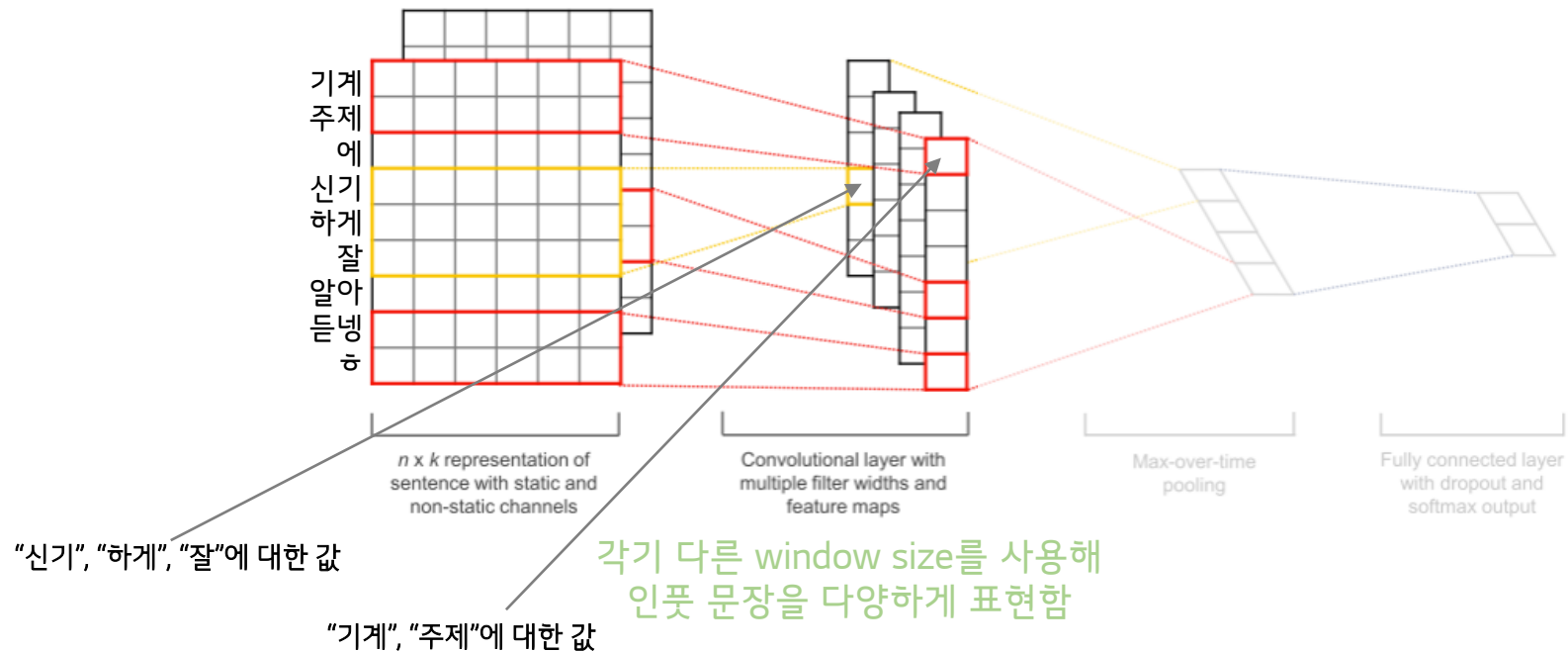
## CNN Step by Step

## 1) INPUT

: 시퀀스 길이  $\times$  단어 임베딩  
차원의 매트릭스

## 2) Convolution

:  $n$ 개 토큰씩 convolution을  
적용하여 feature 생성



## CNN Step by Step

## 1) INPUT

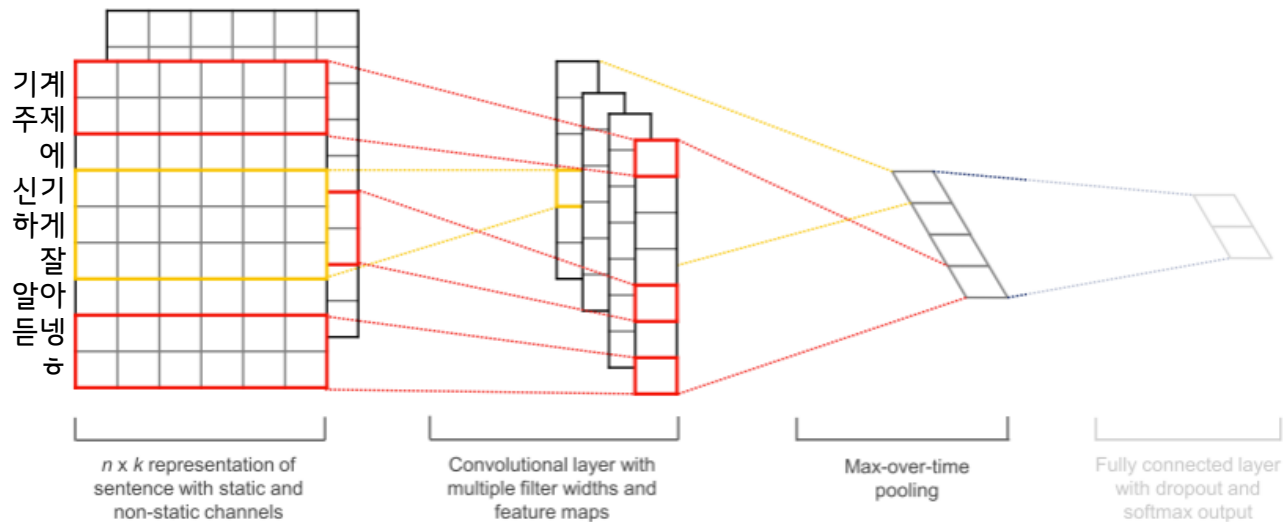
: 시퀀스 길이  $\times$  단어 임베딩  
차원의 매트릭스

## 2) Convolution

:  $n$ 개 토큰씩 convolution을  
적용하여 feature 생성

## 3) Max Pooling

: 각 feature map에서 max  
를 뽑아오는 연산 진행



**\*\* Max Pooling을 통해 하나의 필터당 하나의 feature만이 남고, 인풋의 길이와 상관 없이 처리가 가능해짐.**

## CNN Step by Step

## 1) INPUT

: 시퀀스 길이  $\times$  단어 임베딩  
차원의 매트릭스

## 2) Convolution

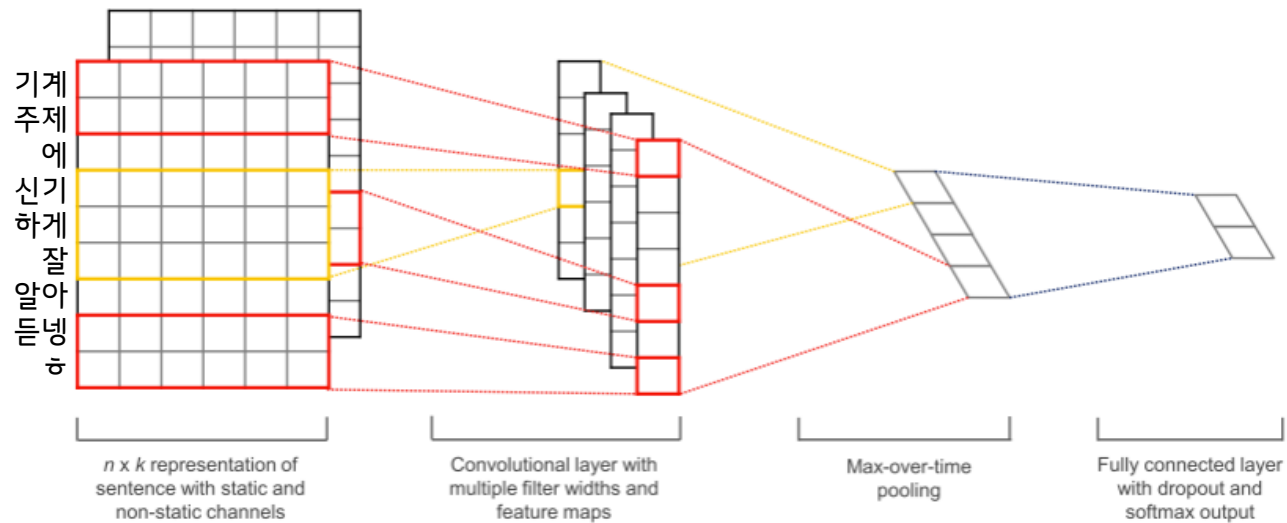
:  $n$ 개 토큰씩 convolution을  
적용하여 feature 생성

## 3) Max Pooling

: 각 feature map에서 max  
를 뽑아오는 연산 진행

## 4) FC Layer

: 태스크를 수행할 수  
있는 FC layer 연결





## NLU 태스크 적용 실험

Convolutional Neural Networks for Sentence Classification (Yoon Kim, 2014)

Data	클래스 개수	데이터 수	단어 개수	정확도	설명
MR	2	10,662	18,765	81.5%	영화 리뷰 긍정/부정 분류
STT-1	5	11,855	17,836	48.0%	리뷰를 매우긍정/긍정/중립/부정/매우부정 분류
STT-2	2	9,613	16,185	87.2%	SST1 데이터를 긍정/부정만 분류하는 태스크
Subj	2	10,000	21,323	93.4%	문장이 주관적인지 객관적인지 분류하는 태스크
TREC	6	5952	9,592	93.6%	질문을 사람/ 지역/ 숫자/ 정보 등 6가지로 분류
CR	2	3775	5,340	84.3%	제품의 Customer Review를 긍정/부정 분류

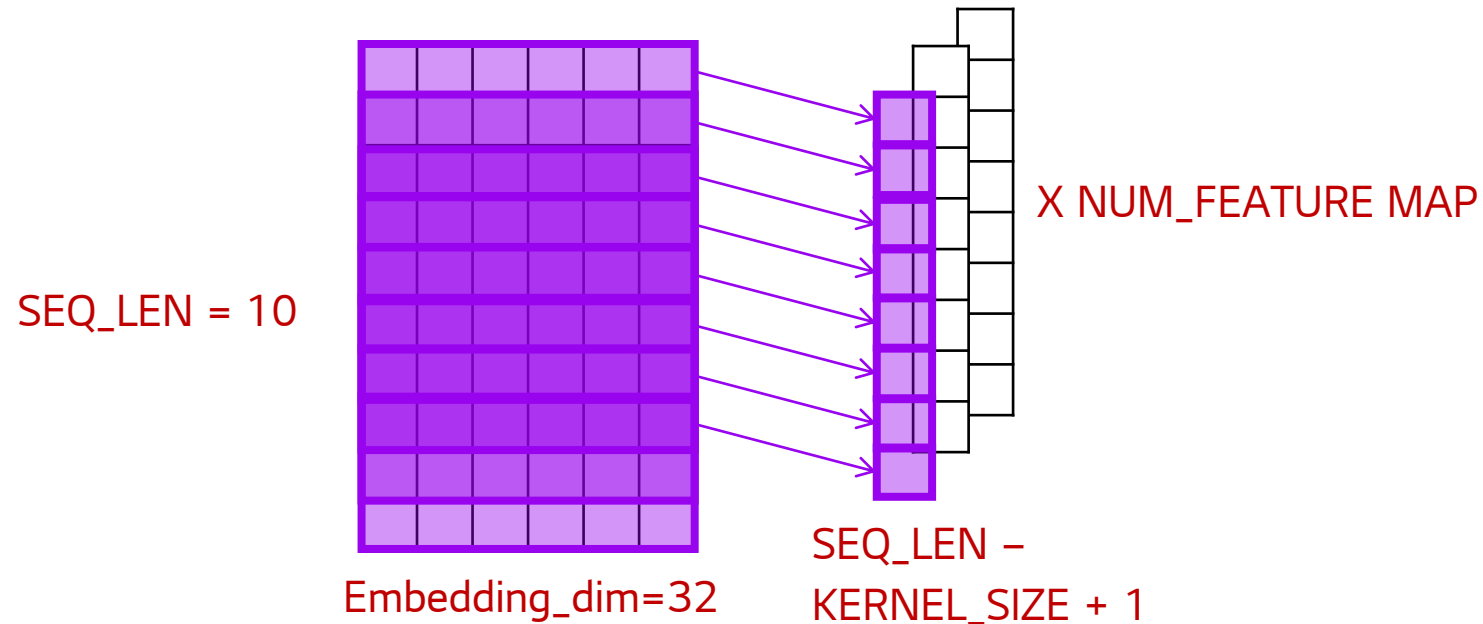
- CNN 필터 윈도우 크기 = 3, 4, 5
- CNN feature map 사이즈 = 각각 100 feature 생성

## NLU를 위한 CNN 연산과 친해지기

미니 실습3\_CNN과 친해지기.ipynb

```
KERNEL_SIZE = 3
NUM_FEATURE_MAP = 10

cnn_applied = Conv1D(filters = NUM_FEATURE_MAP,
                      kernel_size = KERNEL_SIZE,
                      activation='relu')(embedded_input)
```

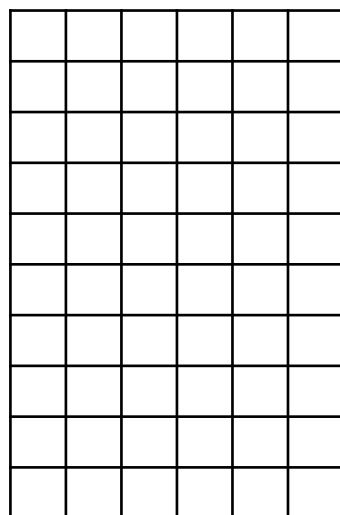


## NLU를 위한 CNN 연산과 친해지기

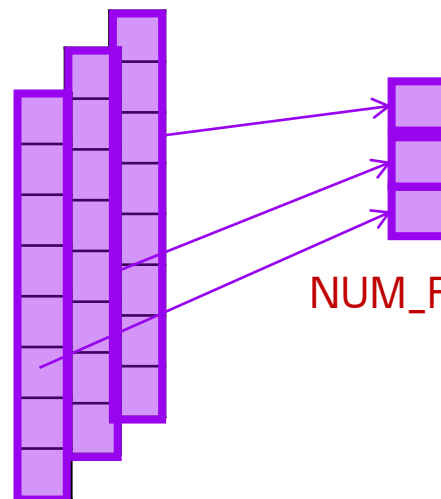
```
P00L_SIZE1 = SEQ_LEN - KERNEL_SIZE1 + 1
print("CNN applied size for kernel {} => {}".format(KERNEL_SIZE1, cnn_applied1.shape))
print("Pooling size for kernel {}      => {}".format(KERNEL_SIZE1, P00L_SIZE1))

max_pool_applied1 = MaxPool1D(pool_size = P00L_SIZE1)(cnn_applied1)
print(max_pool_applied1.shape)
```

SEQ\_LEN = 10



Embedding\_dim=32

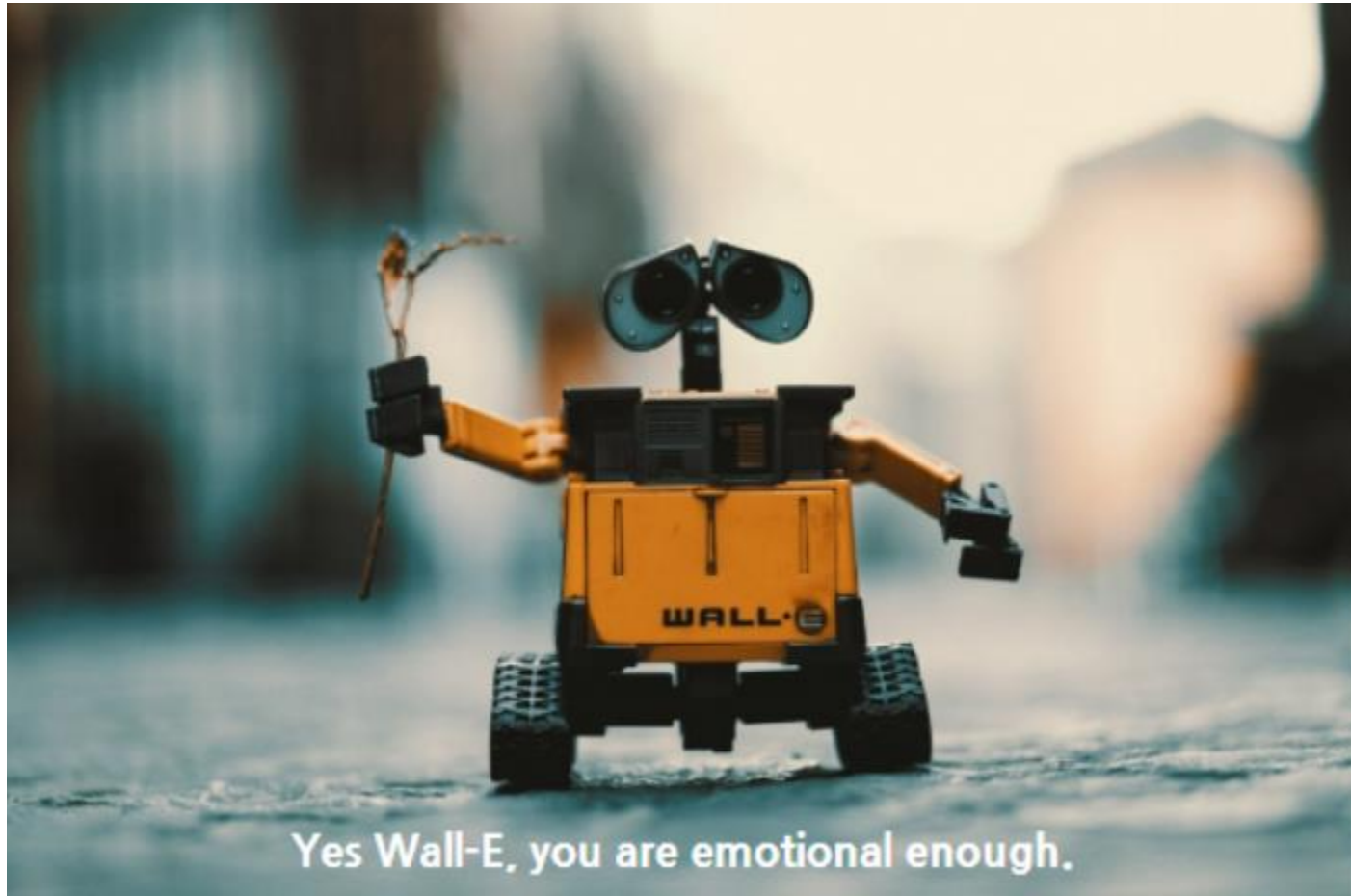


NUM\_FEATURE\_MAP

$$\text{SEQ\_LEN} - \text{KERNEL\_SIZE} + 1$$

CNN을 사용하여 감성분석 모델 학습하기

실습\_4\_CNN\_for\_NLU.ipynb



## CNN을 사용하여 감성분석 모델 학습하기

- 데이터 : 네이버 영화리뷰 (<https://github.com/e9t/nsmc>)

## 개봉영화 평점

저 산 너머 ★★★★★ 8.41



어벤져스: 인피니티 워 ★★★★★ 8.96

그렇다. 토르는 신이었다...  
내년까지 살아가야 할 이유가 생겼습니다  
이 영화의 최고의 빌런은 번역가다.

1917 ★★★★★ 8.88

트롤: 월드 투어 ★★★★★ 8.91

- 네이버 영화 평점 크롤링 데이터 20만 건
- 평점 1-4 댓글 = 부정
- 평점 9-10 댓글 = 긍정 으로 라벨링

- 학습 목표 :
  - CNN을 사용한 자연어 분석 모델을 이해한다.
  - TF Keras를 이용해 모델을 구축하고, 학습 및 추론할 수 있다.