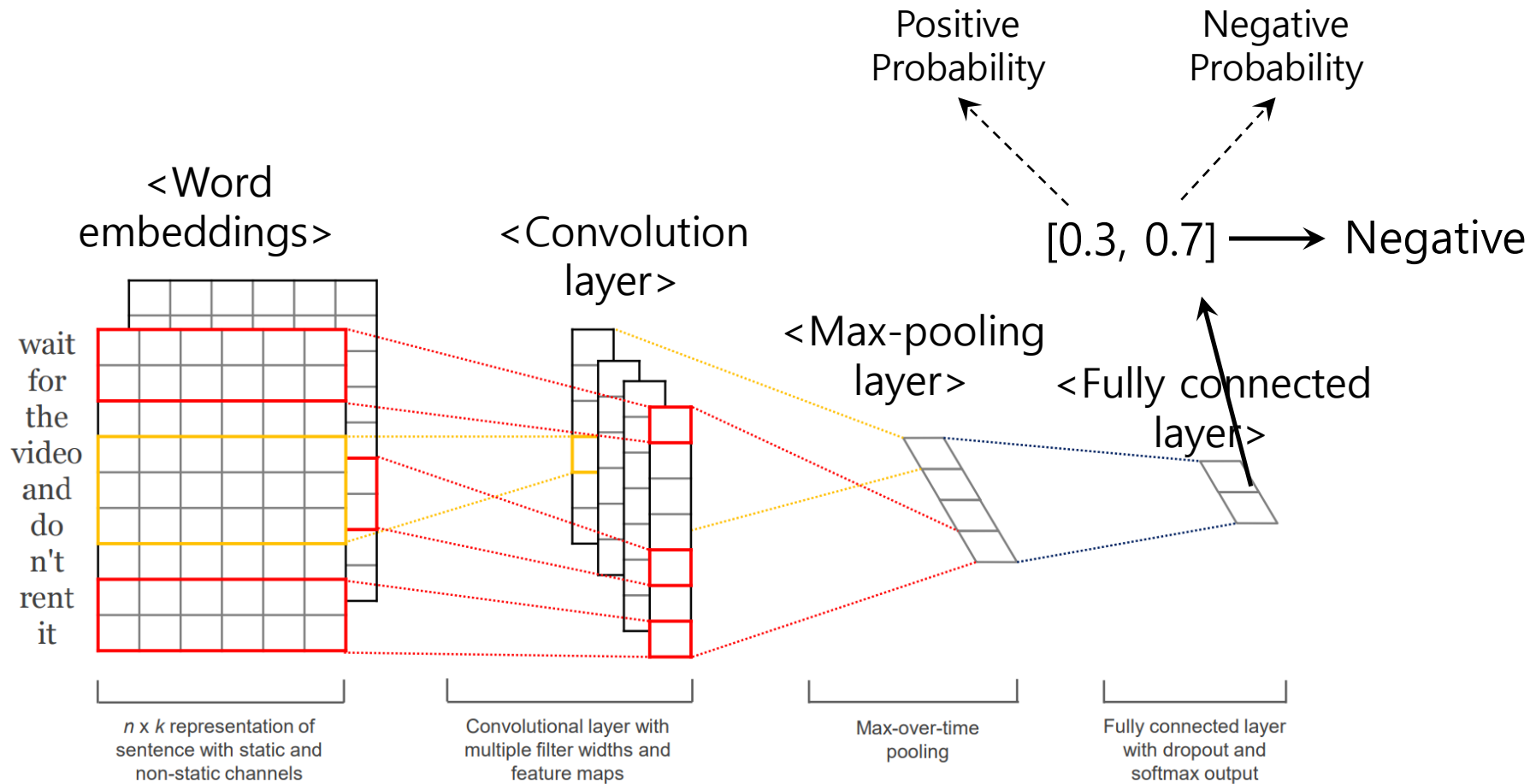
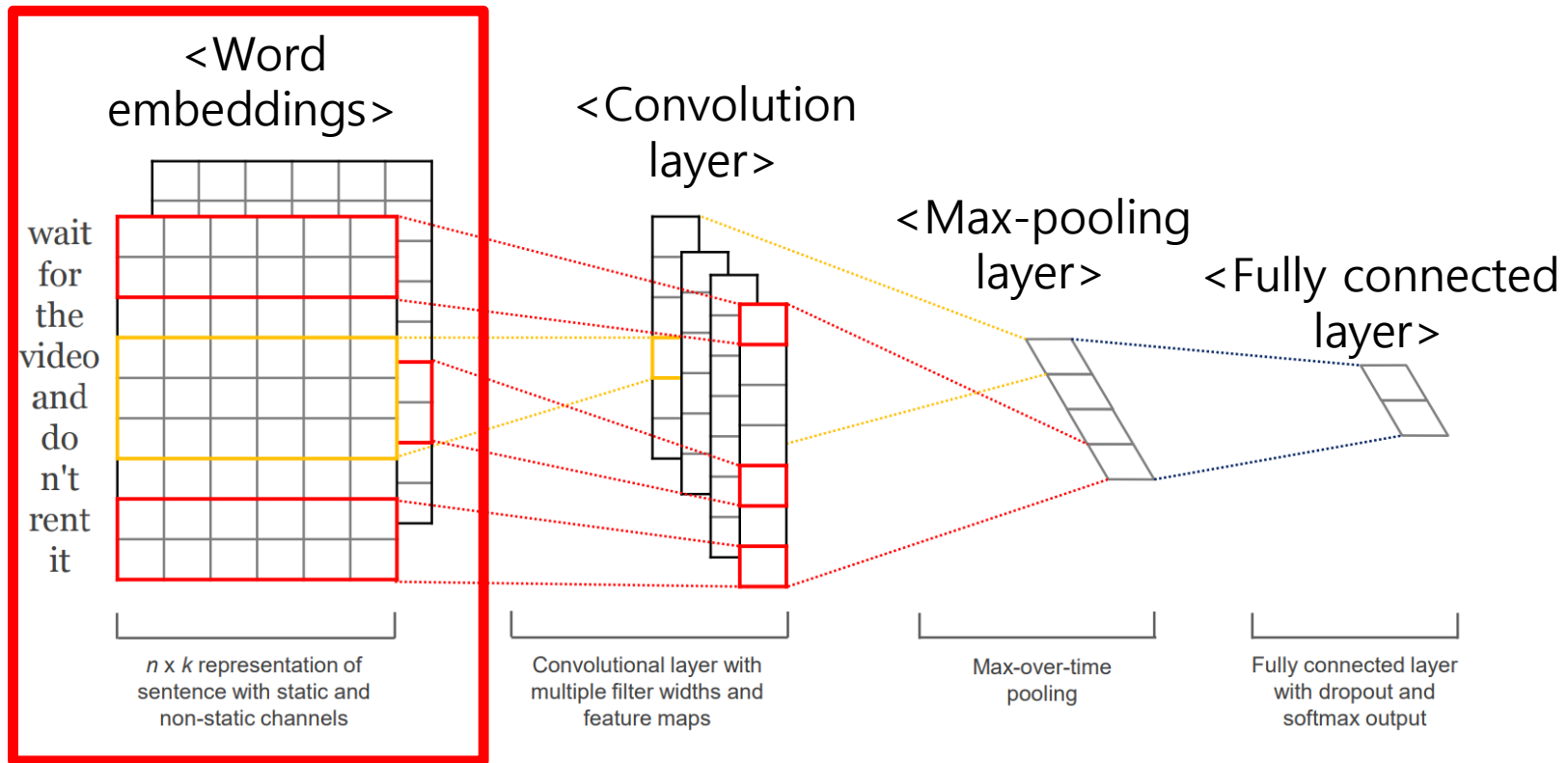


# CNN for sentence classification<sup>[1]</sup>



[1] Kim, Yoon. "Convolutional neural networks for sentence classification." Conference on Empirical Methods in Natural Language Processing (2014).

# CNN for sentence classification<sup>[1]</sup>

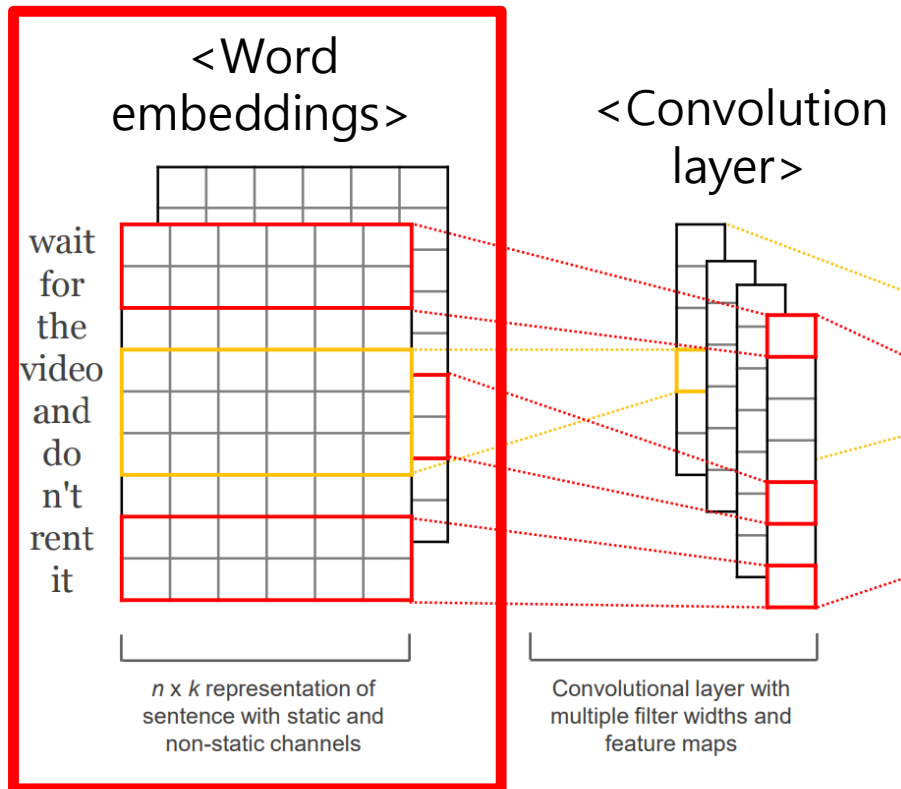


**[n, k] (k=300)**

**<Word embeddings>**

- Used embeddings :  
**word2vec**, pretrained on Google News (Mikolov et.al., 2013)
- Embedding size = 300

# CNN for sentence classification<sup>[1]</sup>

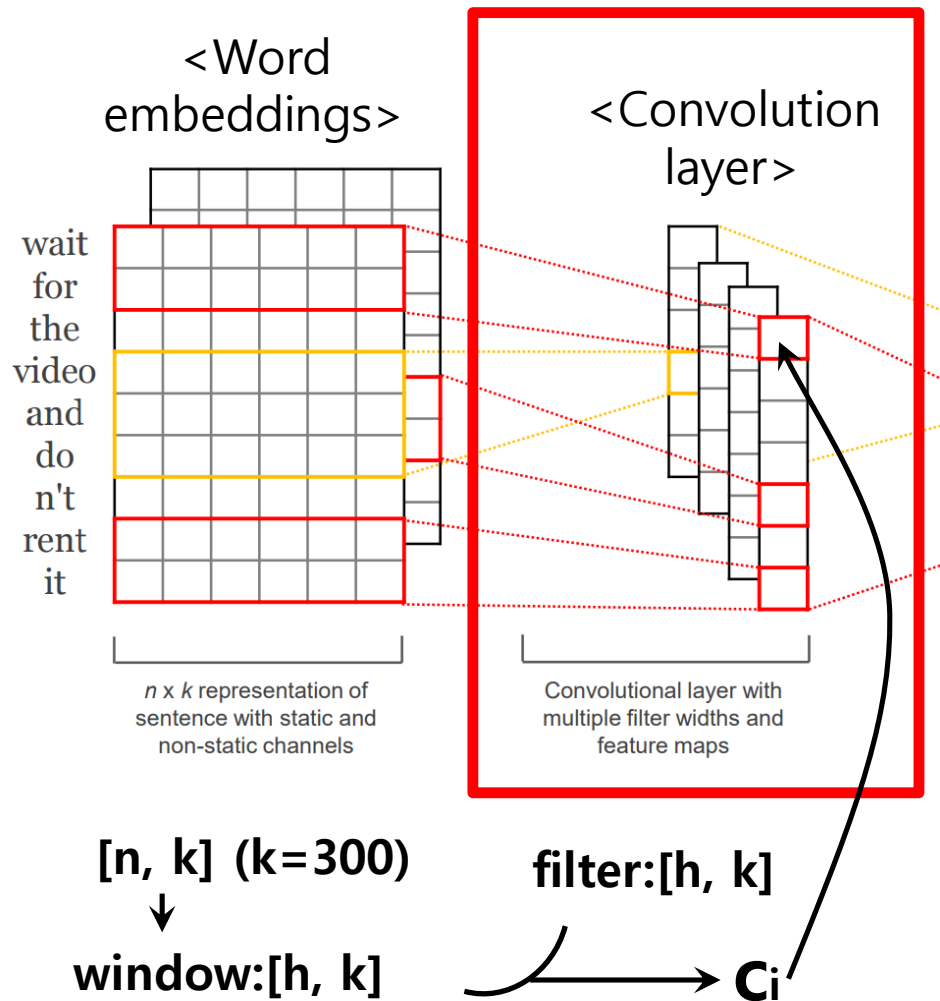


**[n, k] (k=300)**

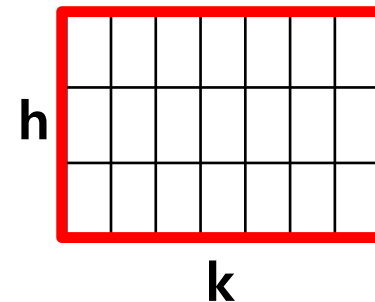
Variations	Embedding Initialization
CNN-rand	X
CNN-non-static	word2vec

- CNN-non-static performed much better compared to CNN-rand. (4%p better in average)
- Proved **initializing embeddings is important.**

# CNN for sentence classification<sup>[1]</sup>

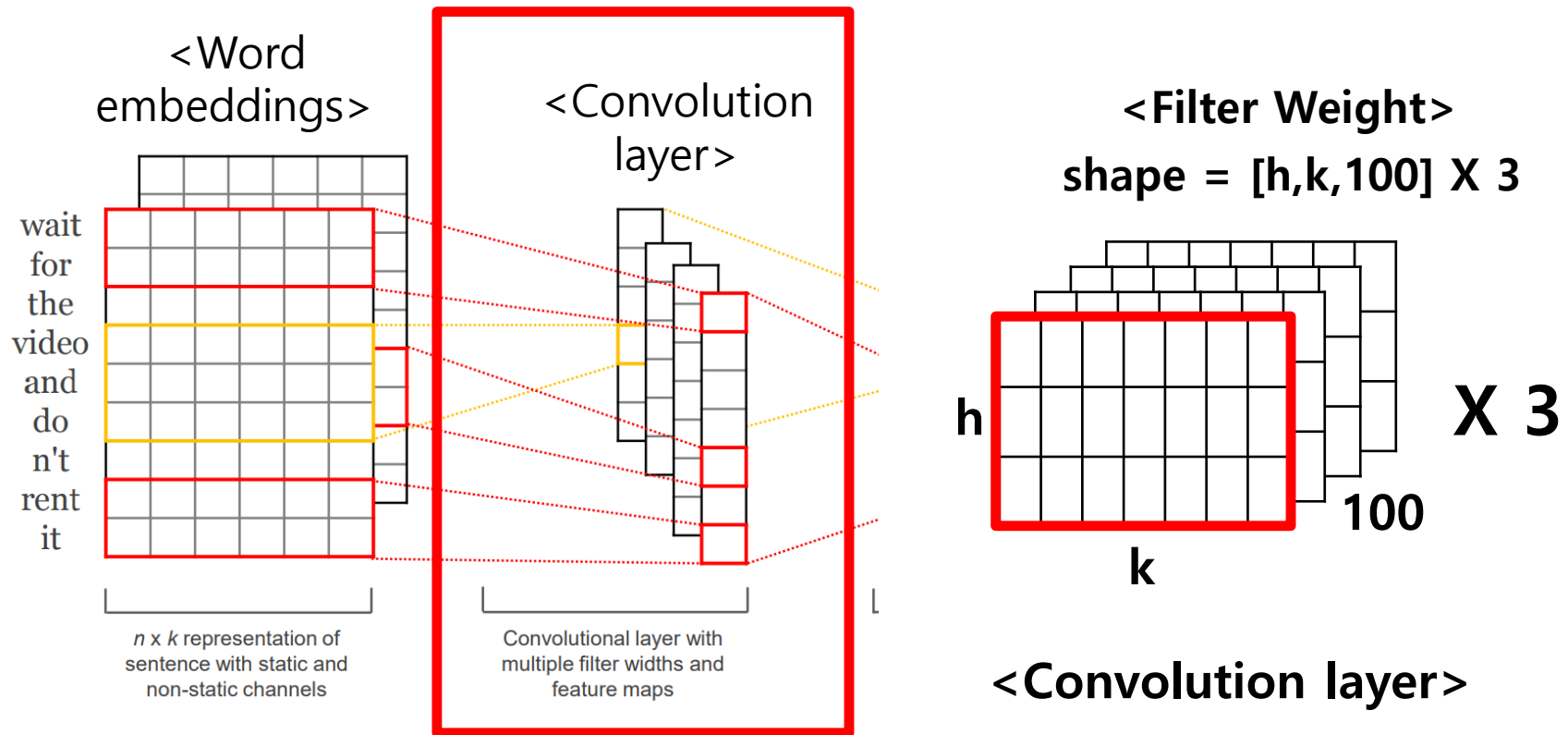


**<Filter Weight>**  
**shape =  $[h, k]$**



- Convolution through filters **can preserve important n-gram feature.**

# CNN for sentence classification<sup>[1]</sup>

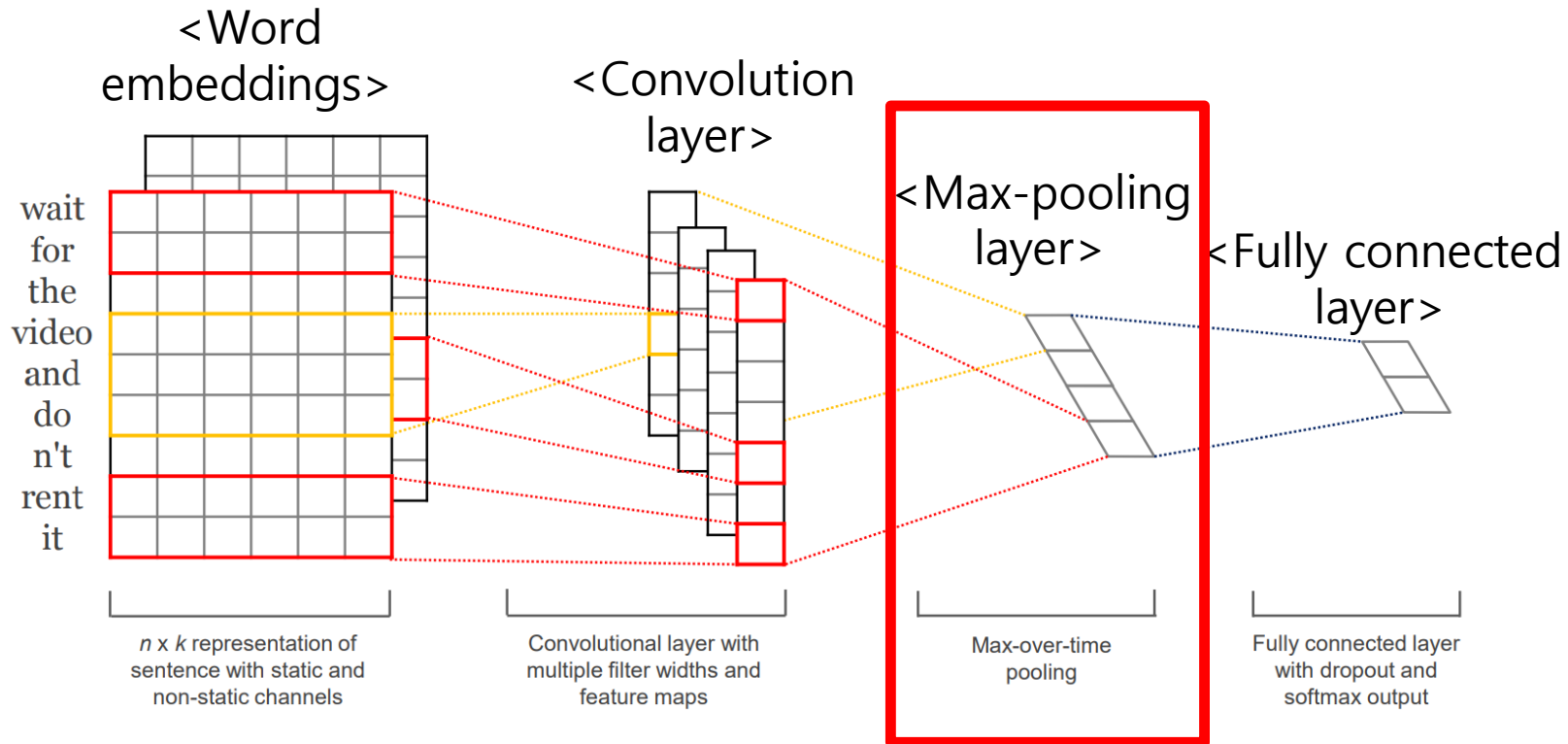


$$c_i = f(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b)$$

$$\mathbf{c} = [c_1, c_2, \dots, c_{n-h+1}]$$

- filter windows( $h$ ) : 3, 4, 5
- 100 filters for each  $h$  (total 300 filters)

# CNN for sentence classification<sup>[1]</sup>

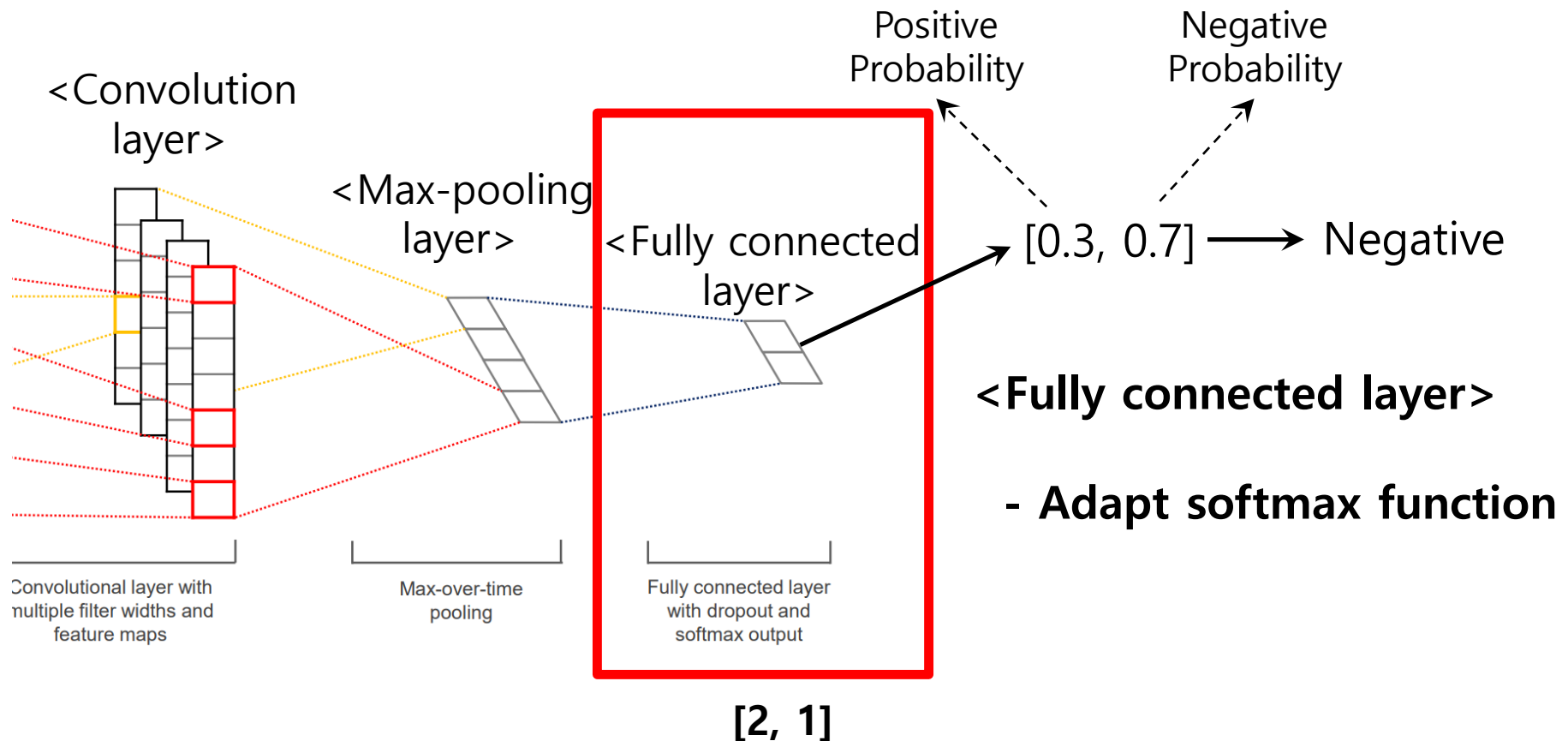


$$\hat{c} = \max\{c\}$$

<Max-pooling layer>

- **Max-over-time-pooling**  
(extract **max value** of each layer)

# CNN for sentence classification<sup>[1]</sup>



# Code review

## **[Objective]**

Implement the CNN models (CNN-rand, CNN-static, CNN-multichannel)

Experiment on two dataset (MR, TREC) - datasets are provided.

Classify sentences into each classes, respectively.

Use provided basic code (CNN-non-static).

## **[PyTorch Code structure]**

- **sentence\_cnn\_train.py**
- **sentence\_cnn.py**
- **sentence\_cnn\_evaluation.py**
- **data\_helpers.py**

## **[TensorFlow Code structure]**

- **text\_cnn\_train.py**
- **text\_cnn.py**
- **text\_cnn\_eval.py**
- **data\_helpers.py**



# Dataset

Data	Description	Target Class
MR (Pang and Lee, 2005)	Movie reviews	2
TREC (Li and Roth, 2002)	Classifying a question into 6 question types	6

The classes in TREC-6 are

- ABBR - Abbreviation
- DESC - Description and abstract concepts
- ENTY - Entities
- HUM - Human beings
- LOC - Locations
- NYM - Numeric values

# Model Variations

Variations	Embedding Initialization	Modified during training	Multichannel
CNN-rand	X	O	X
CNN-static	word2vec	X	X
CNN-non-static	word2vec	O	X
CNN-multichannel	word2vec	O for one, X for the other	O (two channels)

# Experiments

Model	MR	SST-1	SST-2	Subj	TREC
CNN-rand	76.1	45.0	82.7	89.6	91.2
CNN-static	81.0	45.5	86.8	93.0	92.8
CNN-non-static	81.5	48.0	87.2	93.4	93.6
CNN-multichannel	81.1	47.4	88.1	93.2	92.2

# Assignment 8: CNN for sentence classification

## [Evaluation report]

Convolutional Neural Network for Sentence Classification-Evaluation Report												
	Accuracy		Hyperparameters									
	MR	TREC	Activation Function	Filter sizes	# Filter	Dropout rate	l2 constraint	lr decay	Optimizer	Learning Rate	Embedding dimension size	mini-batch size
CNN-rand			ReLU	3,4,5	100	0.5	0.0001	step-0.99	Adadelta	0.1	300	50
CNN-static												
CNN-non-static												
CNN-multichannel												
CNN-rand												
CNN-static												
CNN-non-static												
CNN-multichannel												
[결과 정리]												
(Validation Dataset Accuracy Plot 첨부하시기 바랍니다)												

- Find the hyperparameter values in the paper,
- If certain hyperparameter is not stated in the paper, then **find proper value** through experiments.

# Assignment 8: CNN for sentence classification

- **Evaluation Criteria**

<b>Simplicity</b>	How concisely did you write the code? - 배점 7점 TREC data preprocessing: +2점 CNN-rand: +1점 CNN-static: +1점 CNN-Multichannel: +2점
<b>Performance</b>	How well did the results of the code perform? - 배점 4점 - MR: acc. 81.5% 이상 달성 2점 - TREC: acc. 92% 이상 달성 2점
<b>Brevity and Clarity</b>	How concisely and clearly did you explain the results? - 배점 4점

## Assignment 8: CNN for sentence classification

- Due to : ~ **11.15(Sun)**
- Submission : Online submission on blackboard
- Your submission should contain
  - 1) The whole code of your implementation
  - 2) The evaluation report
- You must implement the components yourself!
- File name : StudentID\_Name.zip