Deep Learning for Natural Language Processing

Convolutional neural networks

Marco Kuhlmann
Department of Computer and Information Science
Linköping University

WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

Limitations of the bag-of-words representation

- The bag-of-words representation is unable to account for meaning that emerges from interactions between words.

  not pleasant, *hardly a generous offer*

- One approach to overcoming this limitation is to introduce explicit features for word pairs, triplets, and longer n-grams.

- However, this would result in very large embedding matrices, and suffer from data sparsity.

  The n-grams *quite good* and *very good* would be completely independent.

Convolutional neural networks

- **Convolutional neural networks (CNNs)** learn to extract *n*-gram features from a text.
- In a first step, a set of *filter functions* transform a sequence of embeddings into a matrix.

  filter = parameterised convolution + non-linear transfer function
- In a second step, a *pooling operation* reduces this matrix to a single vector, which we can view as a summary of the text.

The convolution operation

$$X * K = \sum_i \sum_j \left(X_{ij} \cdot K_{ij} \right)$$

0.08	0.95	0.85
0.98	0.78	0.02
0.32	0.13	0.82

input X

0.50	0.10	0.10
1.00	0.20	0.20
0.50	0.10	0.10

kernel K

The convolution operation

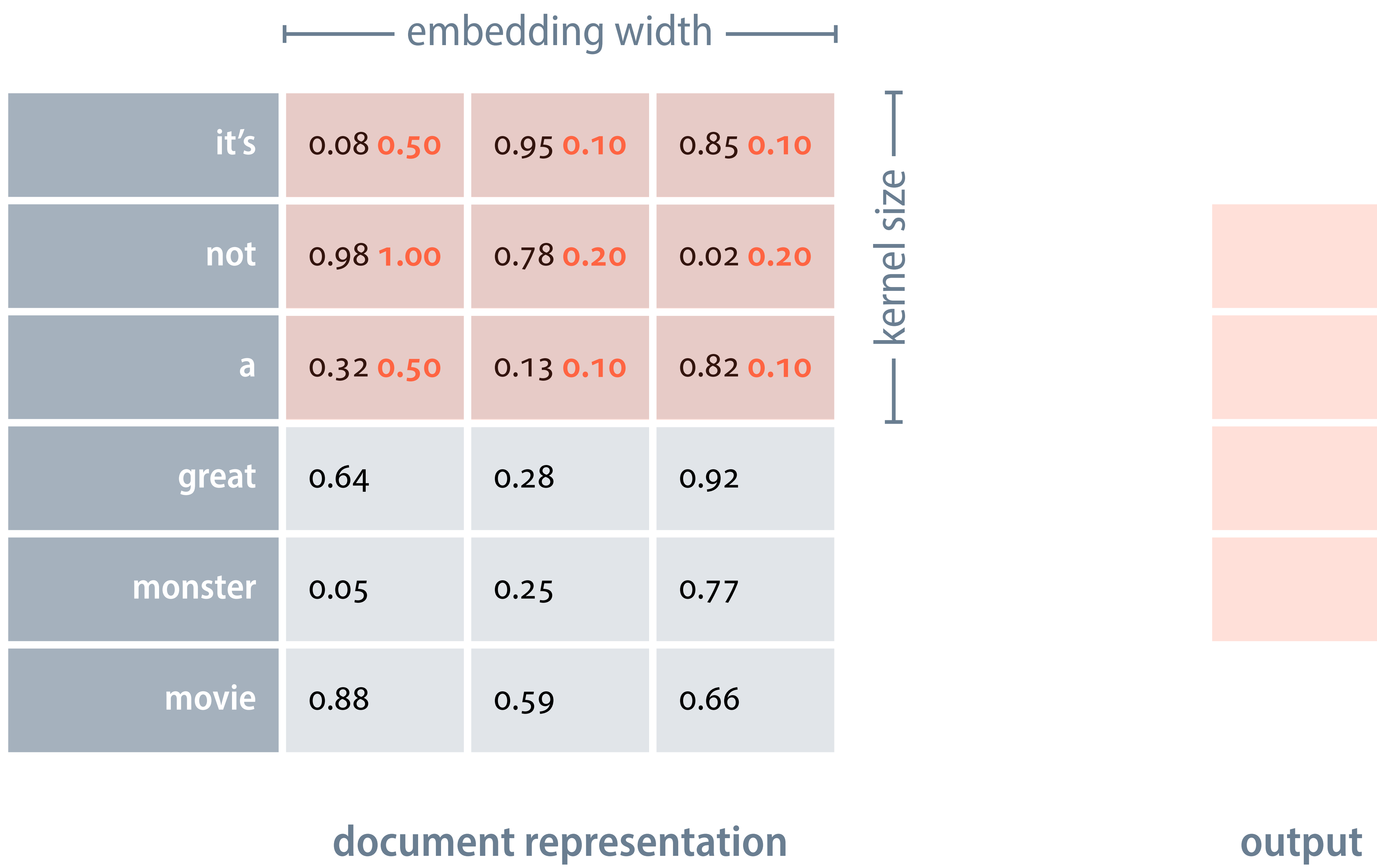
$$X * K = \sum_i \sum_j \left(X_{ij} \cdot K_{ij} \right)$$

0.08 **0.50**	0.95 **0.10**	0.85 **0.10**
0.98 **1.00**	0.78 **0.20**	0.02 **0.20**
0.32 **0.50**	0.13 **0.10**	0.82 **0.10**

convolution

1.615

One-dimensional convolutions over text



One-dimensional convolutions over text

it's	0.08 **0.50**	0.95 **0.10**	0.85 **0.10**
not	0.98 **1.00**	0.78 **0.20**	0.02 **0.20**
a	0.32 **0.50**	0.13 **0.10**	0.82 **0.10**
great	0.64	0.28	0.92
monster	0.05	0.25	0.77
movie	0.88	0.59	0.66

— convolution → 1.615

document representation

output

One-dimensional convolutions over text

One-dimensional convolutions over text

it's	0.08	0.95	0.85
not	0.98	0.78	0.02
a	0.32 0.50	0.13 0.10	0.82 0.10
great	0.64 1.00	0.28 0.20	0.92 0.20
monster	0.05 0.50	0.25 0.10	0.77 0.10
movie	0.88	0.59	0.66

document representation

1.615
1.520
1.262

output

One-dimensional convolutions over text

it's	0.08	0.95	0.85	
not	0.98	0.78	0.02	1.615
a	0.32	0.13	0.82	1.520
great	0.64 **0.50**	0.28 **0.10**	0.92 **0.10**	1.262
monster	0.05 **1.00**	0.25 **0.20**	0.77 **0.20**	1.259
movie	0.88 **0.50**	0.59 **0.10**	0.66 **0.10**	

document representation **output**

Narrow convolutions versus wide convolutions

- When using kernels of size two or more, the output of a convolution will be smaller than the input.

  narrow convolution

- To produce output of the same size as the input, we may pad the input; this is typically done symmetrically on both ends.

  wide convolution

Wide convolutions

<pad>	0.00 **0.50**	0.00 **0.10**	0.00 **0.10**	
it's	0.08 **1.00**	0.95 **0.20**	0.85 **0.20**	1.010
not	0.98 **0.50**	0.78 **0.10**	0.02 **0.10**	1.615
a	0.32	0.13	0.82	1.520
great	0.64	0.28	0.92	1.262
monster	0.05	0.25	0.77	1.259
movie	0.88	0.59	0.66	1.257
<pad>	0.00	0.00	0.00	

Multiple channels

- We can view the dimensions of the word embeddings as separate **input channels**, similar to the colour channels in images.
- By applying several different filters to the same input, we can also get multiple **output channels**.

  several feature maps for the same input
- In deep learning libraries such as PyTorch, we can match input channels to output channels in flexible ways.

  documentation for nn.Conv1d

Multiple channels

<pad>	0.00 **0.50**	0.00 **0.10**	0.00 **0.10**			
it’s	0.08 **1.00**	0.95 **0.20**	0.85 **0.20**	1.010		
not	0.98 **0.50**	0.78 **0.10**	0.02 **0.10**	1.615		
a	0.32	0.13	0.82	1.520		
great	0.64	0.28	0.92	1.262		
monster	0.05	0.25	0.77	1.259		
movie	0.88	0.59	0.66	1.257		
<pad>	0.00	0.00	0.00			

Multiple channels

<pad>	0.00 **0.10**	0.00 **0.50**	0.00 **0.10**
it's	0.08 **0.20**	0.95 **1.00**	0.85 **0.20**
not	0.98 **0.10**	0.78 **0.50**	0.02 **0.10**
a	0.32	0.13	0.82
great	0.64	0.28	0.92
monster	0.05	0.25	0.77
movie	0.88	0.59	0.66
<pad>	0.00	0.00	0.00

1.010	1.626	
1.615	1.727	
1.520	1.144	
1.262	0.978	
1.259	1.159	
1.257	1.050	

Multiple channels

<pad>	0.00 **0.10**	0.00 **0.10**	0.00 **0.50**
it's	0.08 **0.20**	0.95 **0.20**	0.85 **1.00**
not	0.98 **0.10**	0.78 **0.10**	0.02 **0.50**
a	0.32	0.13	0.82
great	0.64	0.28	0.92
monster	0.05	0.25	0.77
movie	0.88	0.59	0.66
<pad>	0.00	0.00	0.00

1.010	1.626	1.242
1.615	1.727	1.355
1.520	1.144	1.648
1.262	0.978	1.974
1.259	1.159	1.859
1.257	1.050	1.369

Pooling

- **Pooling** reduces a feature map to a single scalar by an operation such as taking the maximum or the average.

  max pooling, average pooling

- When performed on all channels, this gives us a single vector representing the entire input document.

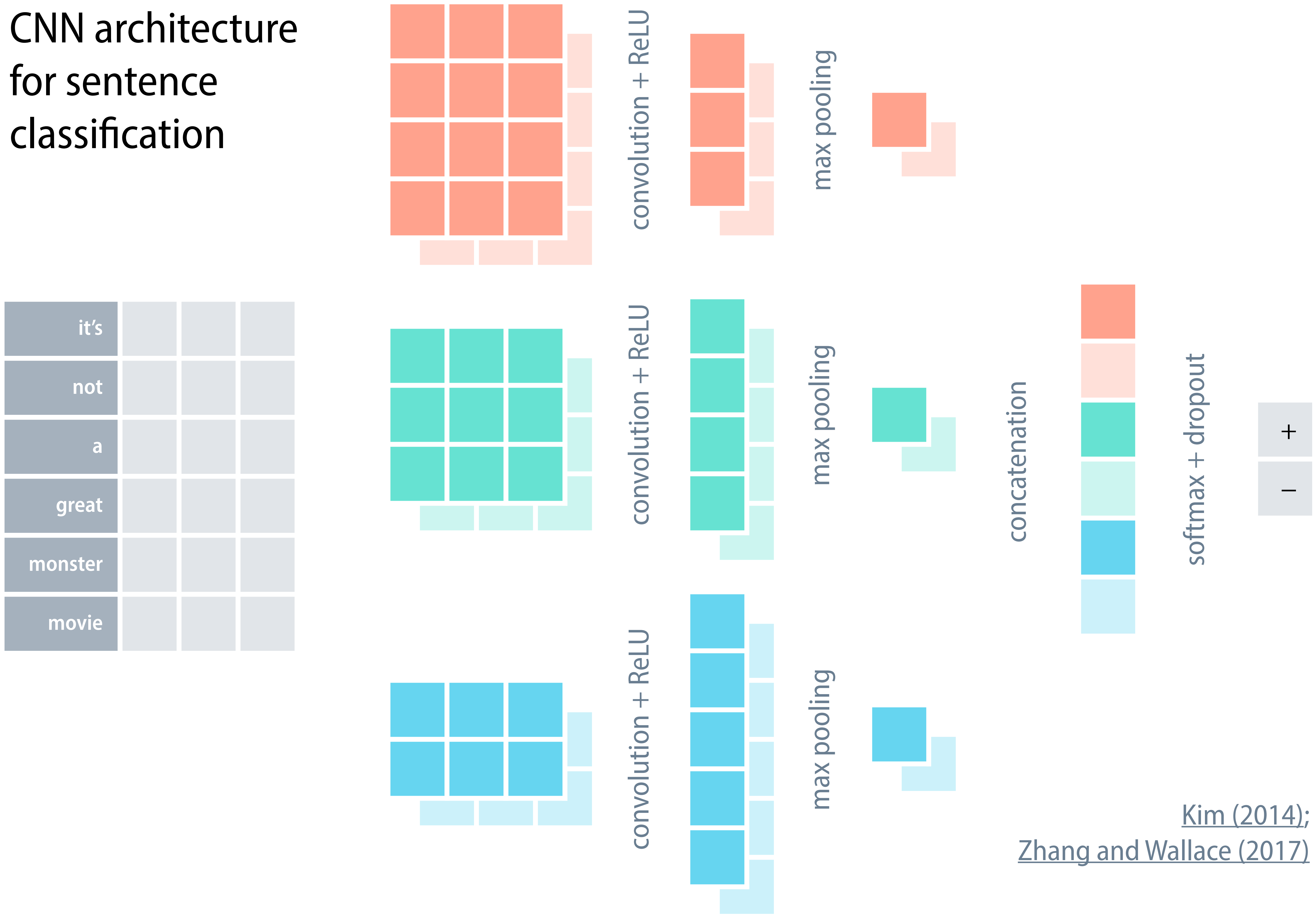
- The hope is that this vector summarises the most important information for the classification task at hand.

Max-pooling

<pad>	0.00	0.00	0.00
it's	0.08	0.95	0.85
not	0.98	0.78	0.02
a	0.32	0.13	0.82
great	0.64	0.28	0.92
monster	0.05	0.25	0.77
movie	0.88	0.59	0.66
<pad>	0.00	0.00	0.00

1.010	1.626	1.242
1.615	1.727	1.355
1.520	1.144	1.648
1.262	0.978	1.974
1.259	1.159	1.859
1.257	1.050	1.369
max	*max*	*max*
1.615	1.727	1.974

CNN architecture for sentence classification



Kim (2014);
Zhang and Wallace (2017)