Neural networks for text

What is text?

You can think of text as a sequence of

- Characters
- **Words**
- Phrases and named entities
- Sentences
- Paragraphs
- …

Bag of words way (sparse)

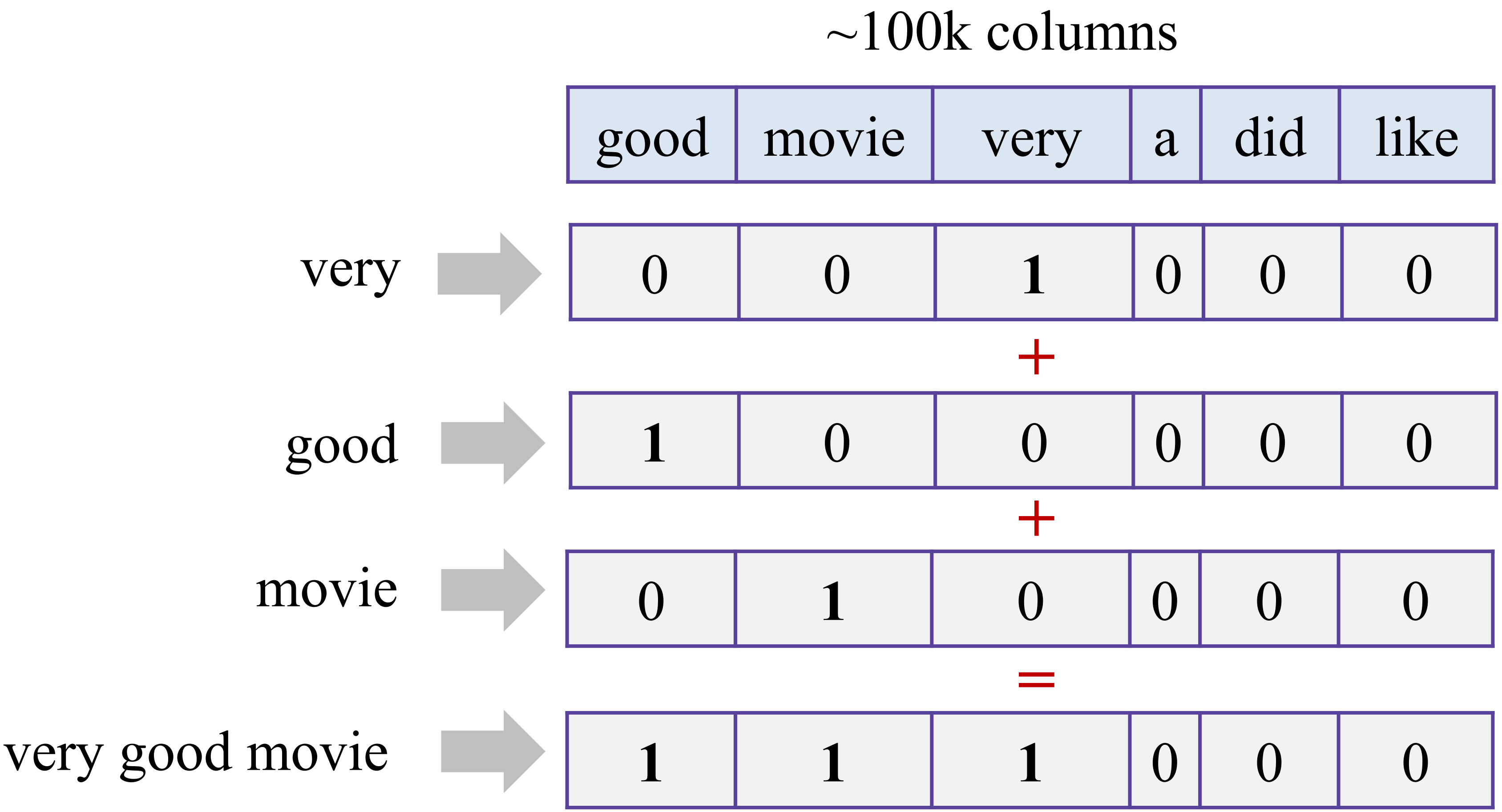
~100k columns

	good	movie	very	a	did	like
very	0	0	**1**	0	0	0
good	**1**	0	0	0	0	0
movie	0	**1**	0	0	0	0

every word with one hot encoded vector.

every word is vectorized independently.

Bag of words way (sparse)

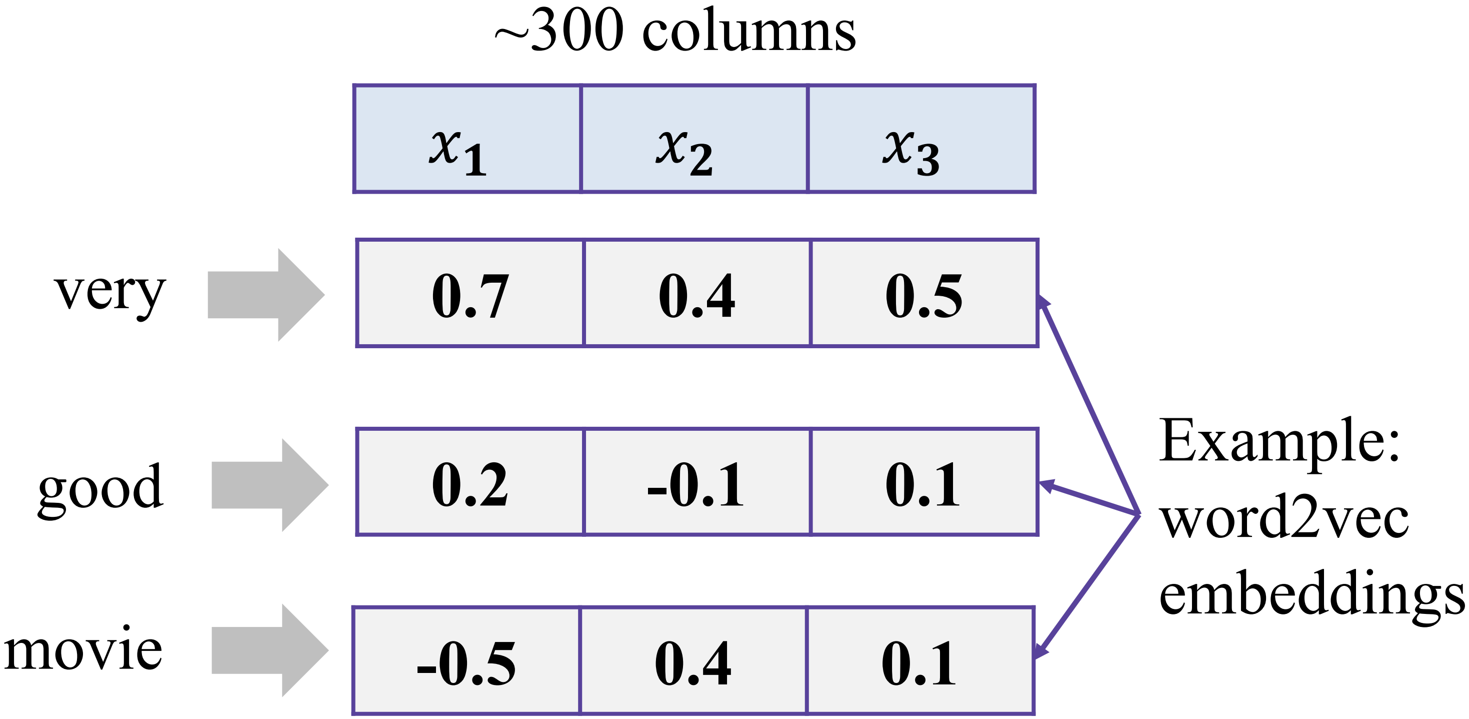


~100k columns

	good	movie	very	a	did	like
very	0	0	**1**	0	0	0
			+			
good	**1**	0	0	0	0	0
			+			
movie	0	**1**	0	0	0	0
			=			
very good movie	**1**	**1**	**1**	0	0	0

Bag of words representation
is a sum of sparse one-hot-encoded vectors

Neural way (dense)



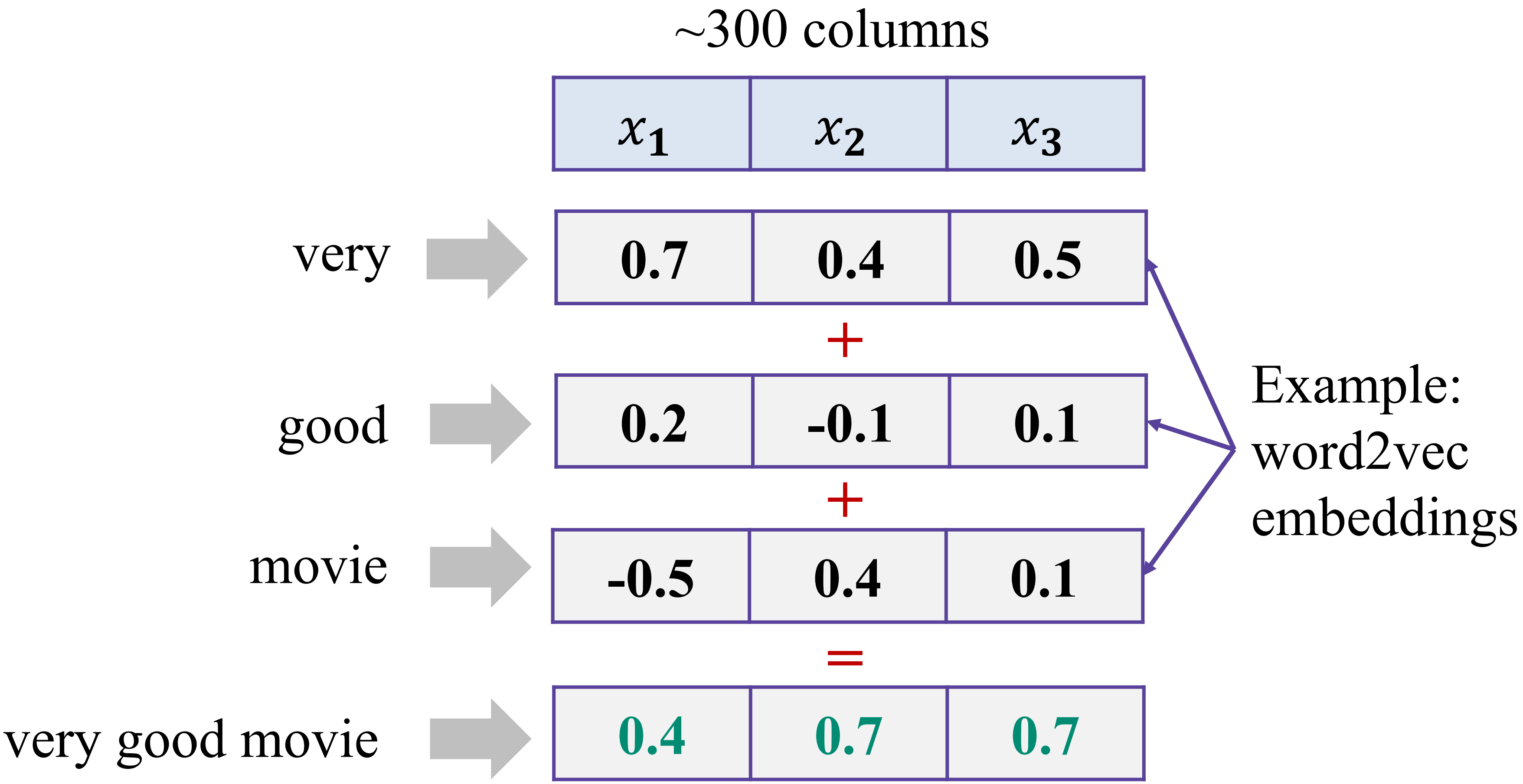
~300 columns

	x_1	x_2	x_3
very	**0.7**	**0.4**	**0.5**
good	**0.2**	**-0.1**	**0.1**
movie	**-0.5**	**0.4**	**0.1**

Example:
word2vec
embeddings

Word2vec property:
Words that have similar context tend to have collinear vectors

Neural way (dense)



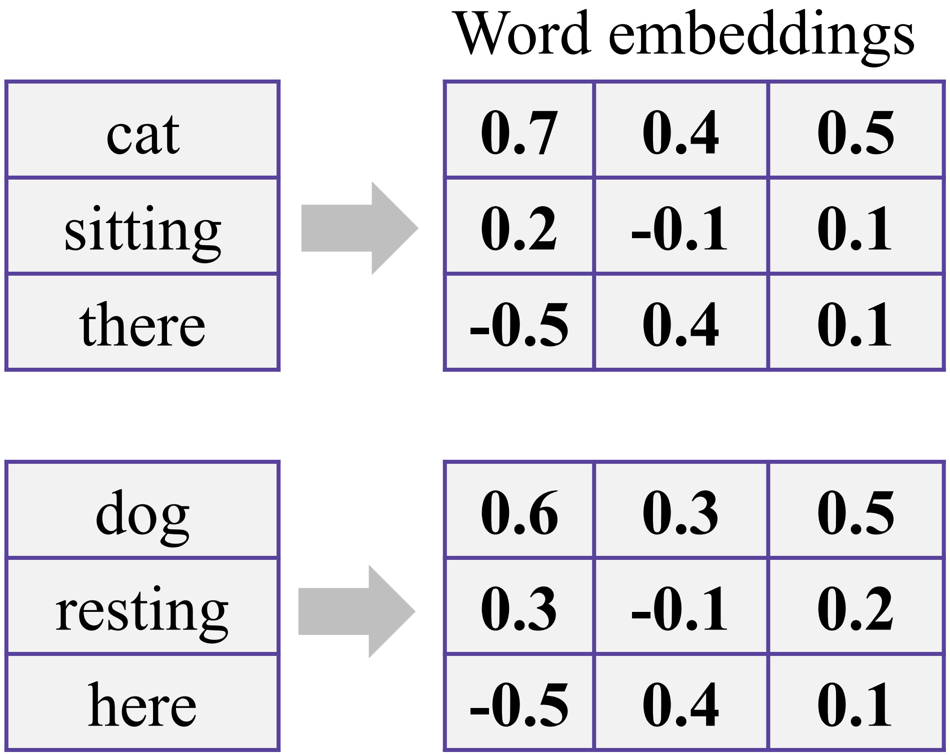
~300 columns

	x_1	x_2	x_3
very	**0.7**	**0.4**	**0.5**
good	**0.2**	**-0.1**	**0.1**
movie	**-0.5**	**0.4**	**0.1**
very good movie	**0.4**	**0.7**	**0.7**

Example:
word2vec
embeddings

Sum of word2vec vectors
can be a good text descriptor already!

A better way: 1D convolutions



Word embeddings

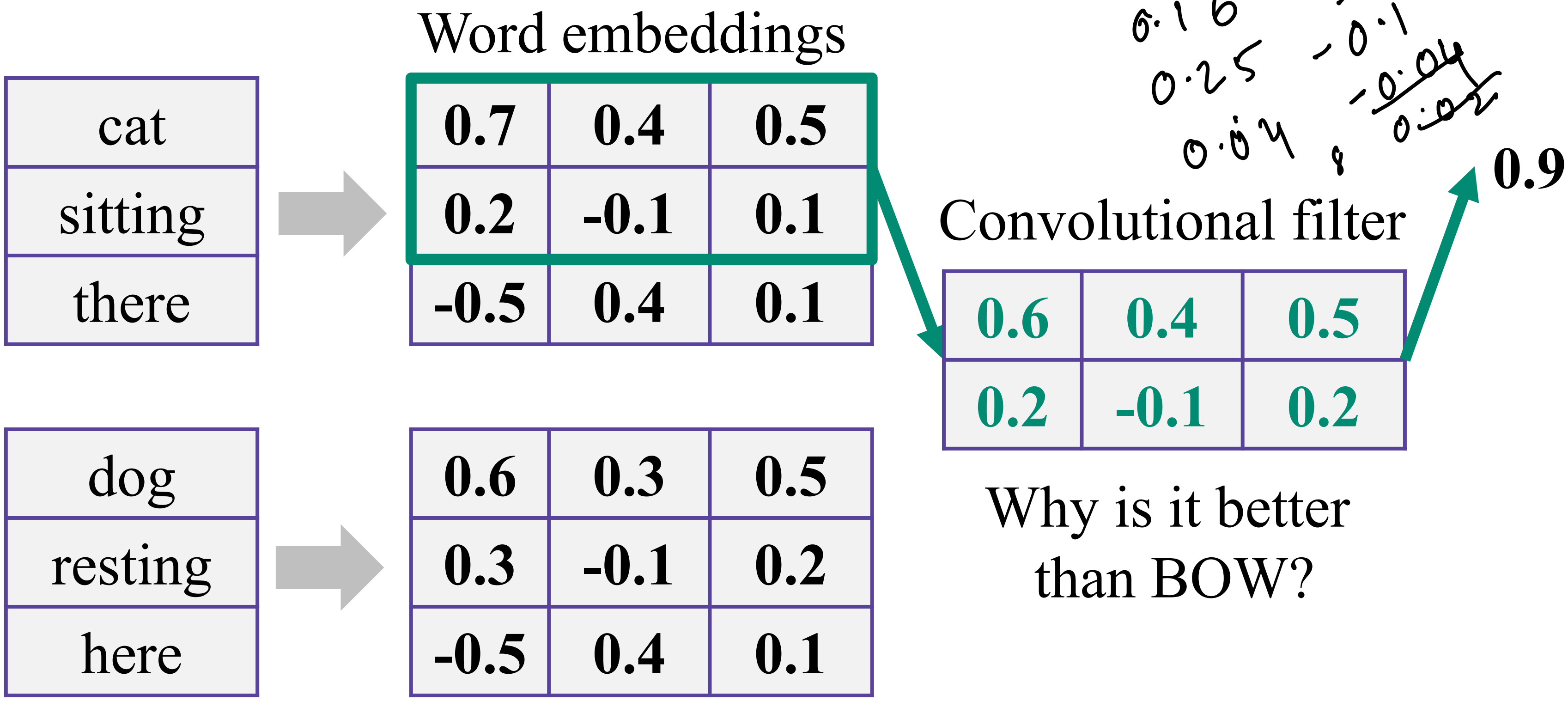
cat	0.7	0.4	0.5
sitting	0.2	-0.1	0.1
there	-0.5	0.4	0.1

dog	0.6	0.3	0.5
resting	0.3	-0.1	0.2
here	-0.5	0.4	0.1

How do we make 2-grams?

http://bionlp-www.utu.fi/wv_demo/

A better way: 1D convolutions



Word embeddings

cat	0.7	0.4	0.5
sitting	0.2	-0.1	0.1
there	-0.5	0.4	0.1

dog	0.6	0.3	0.5
resting	0.3	-0.1	0.2
here	-0.5	0.4	0.1

Convolutional filter

0.6	0.4	0.5
0.2	-0.1	0.2

0.9

0.4^2 0.01

0.16 ~~0.02~~

0.25 -0.1

0.04 ~~-0.04~~

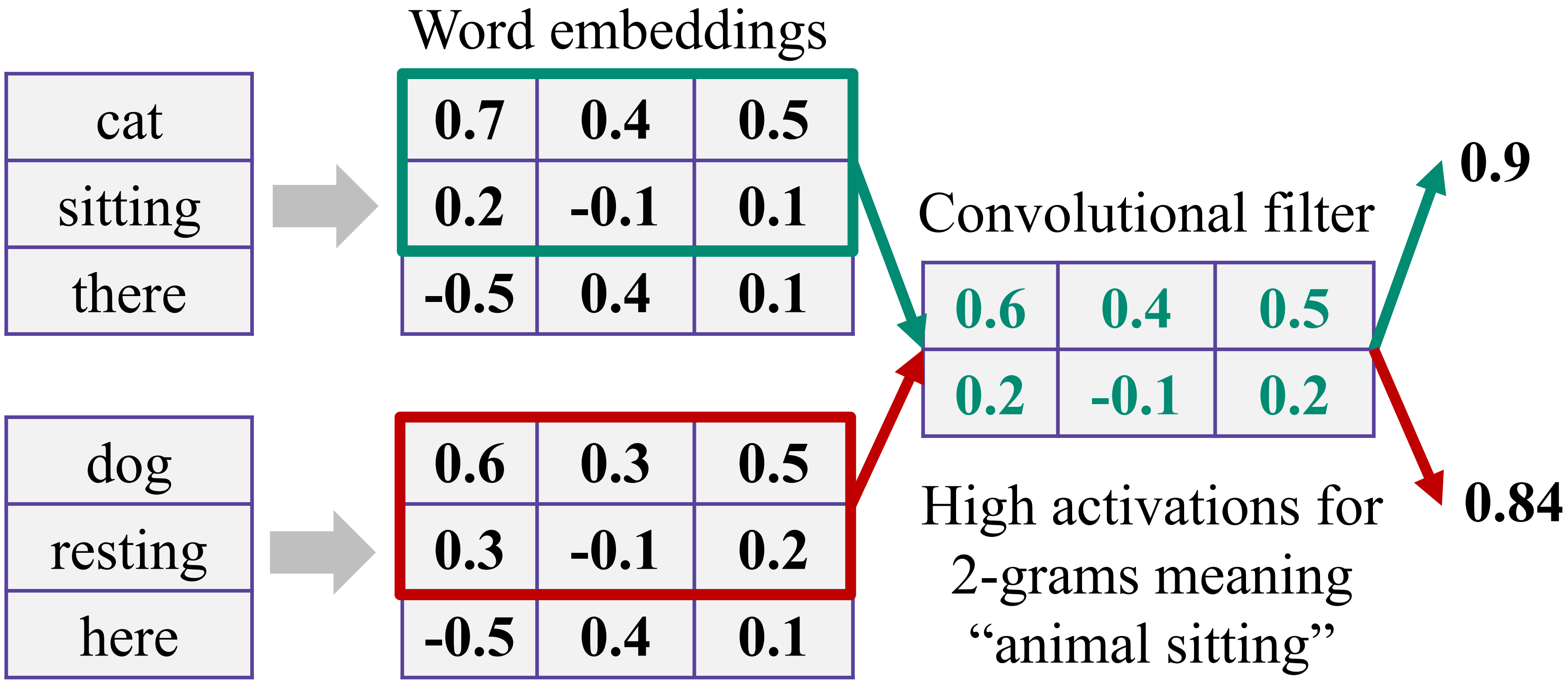
8 ~~0.02~~

Why is it better than BOW?

- This convolution provides high activations for 2-grams with certain meaning

http://bionlp-www.utu.fi/wv_demo/

A better way: 1D convolutions



Word embeddings

cat	0.7	0.4	0.5
sitting	0.2	-0.1	0.1
there	-0.5	0.4	0.1

dog	0.6	0.3	0.5
resting	0.3	-0.1	0.2
here	-0.5	0.4	0.1

Convolutional filter

0.6	0.4	0.5
0.2	-0.1	0.2

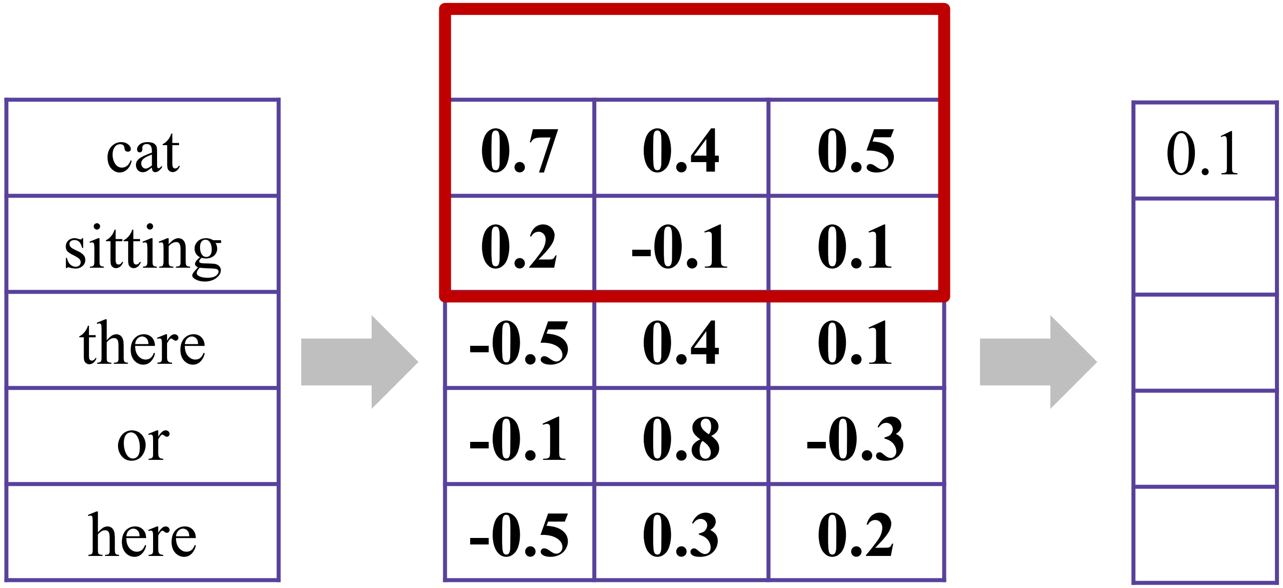
High activations for 2-grams meaning “animal sitting”

- This convolution provides high activations for 2-grams with certain meaning
- Word2vec vectors for similar words are similar in terms of cosine distance (similar to dot product)

http://bionlp-www.utu.fi/wv_demo/

1D convolutions

- Can be extended to 3-grams, 4-grams, etc.
- One filter is not enough, need to track many n-grams
- They are called 1D because we slide the window only in one direction



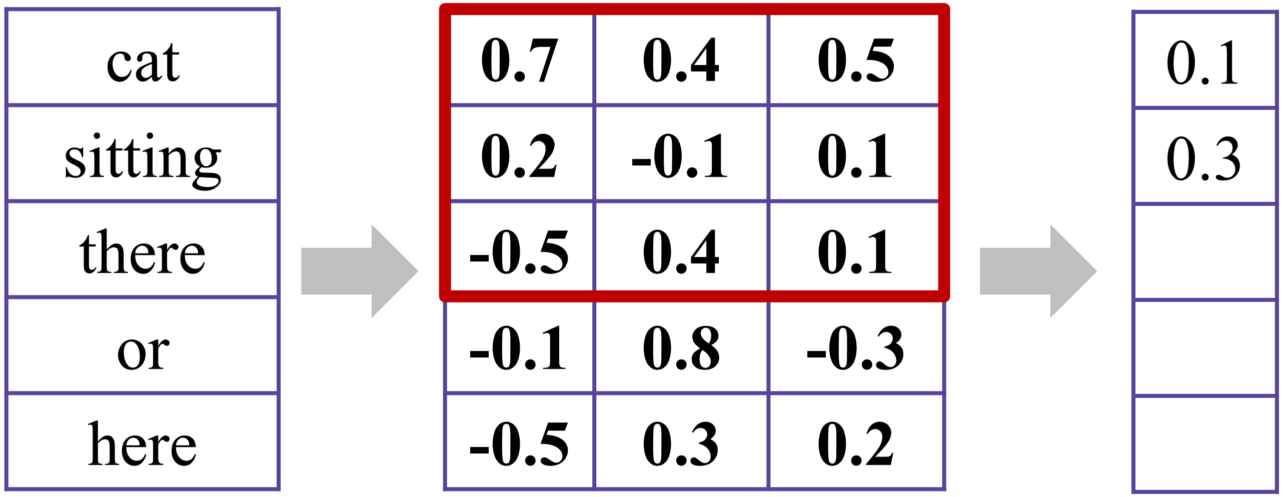
cat
sitting
there
or
here

0.7	**0.4**	**0.5**
0.2	**-0.1**	**0.1**
-0.5	**0.4**	**0.1**
-0.1	**0.8**	**-0.3**
-0.5	**0.3**	**0.2**

0.1

1D convolutions

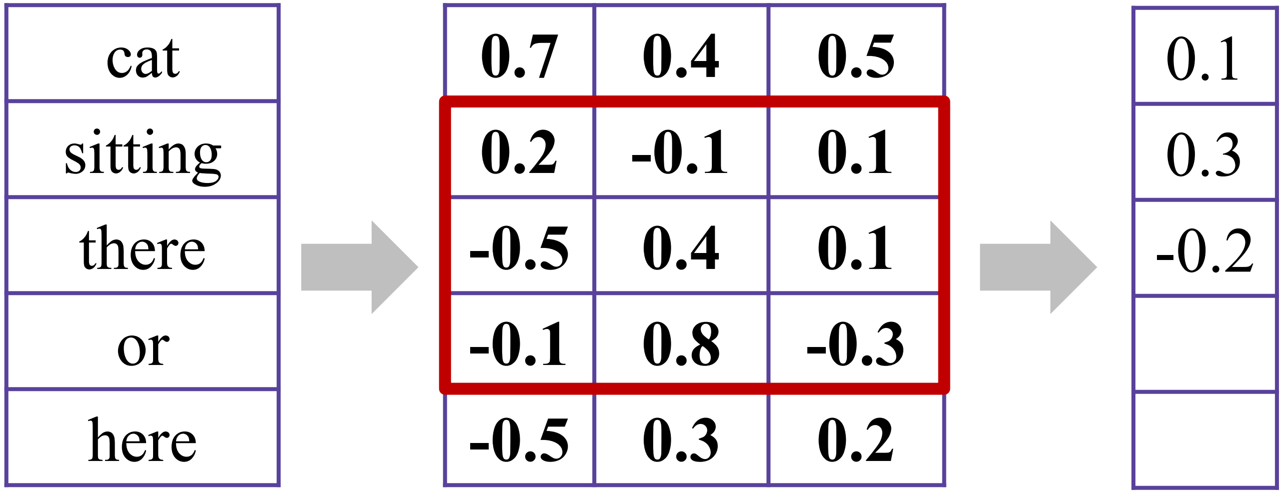
- Can be extended to 3-grams, 4-grams, etc.
- One filter is not enough, need to track many n-grams
- They are called 1D because we slide the window only in one direction



cat	**0.7**	**0.4**	**0.5**	0.1
sitting	**0.2**	**-0.1**	**0.1**	0.3
there	**-0.5**	**0.4**	**0.1**	
or	**-0.1**	**0.8**	**-0.3**	
here	**-0.5**	**0.3**	**0.2**	

1D convolutions

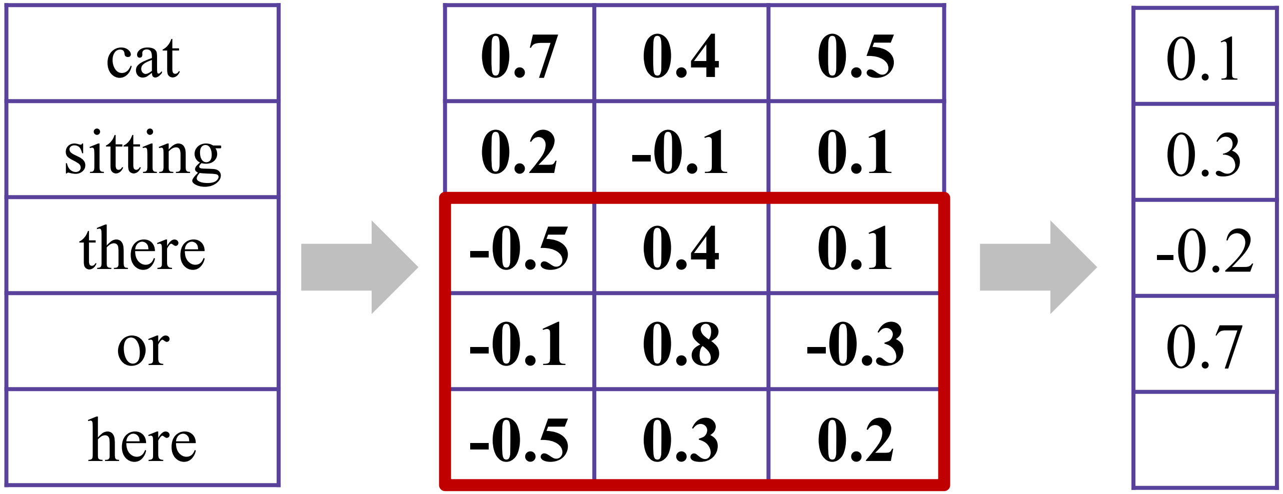
- Can be extended to 3-grams, 4-grams, etc.
- One filter is not enough, need to track many n-grams
- They are called 1D because we slide the window only in one direction



cat	**0.7**	**0.4**	**0.5**		0.1
sitting	**0.2**	**-0.1**	**0.1**		0.3
there	**-0.5**	**0.4**	**0.1**		-0.2
or	**-0.1**	**0.8**	**-0.3**		
here	**-0.5**	**0.3**	**0.2**		

1D convolutions

- Can be extended to 3-grams, 4-grams, etc.
- One filter is not enough, need to track many n-grams
- They are called 1D because we slide the window only in one direction



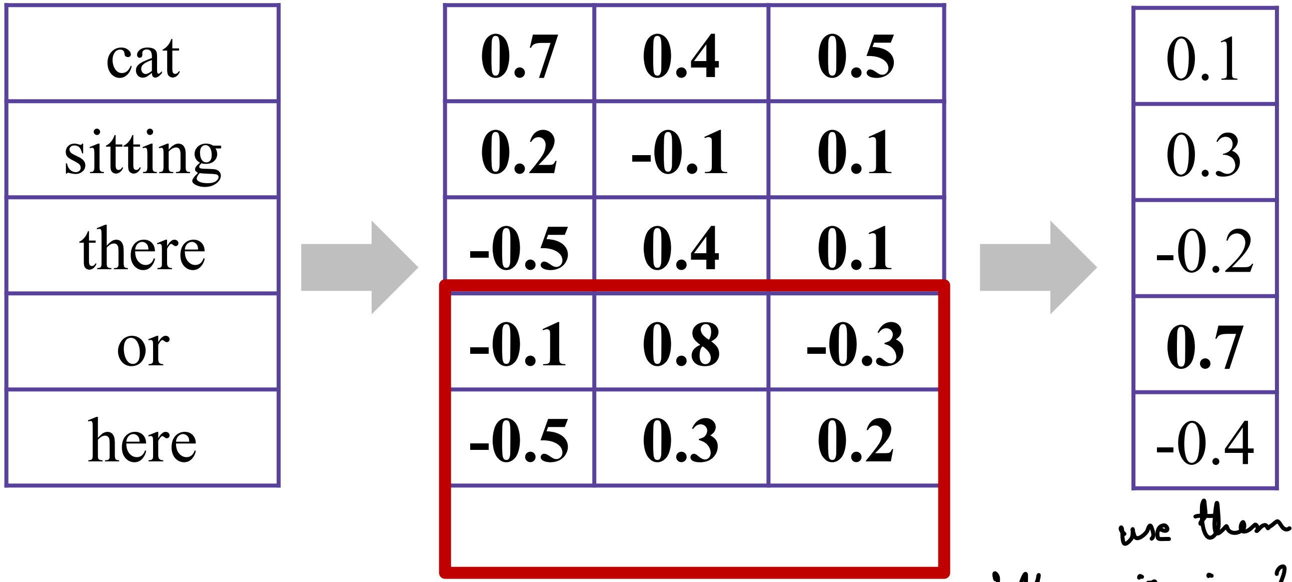
cat
sitting
there
or
here

0.7	**0.4**	**0.5**
0.2	**-0.1**	**0.1**
-0.5	**0.4**	**0.1**
-0.1	**0.8**	**-0.3**
-0.5	**0.3**	**0.2**

0.1
0.3
-0.2
0.7

1D convolutions

- Can be extended to 3-grams, 4-grams, etc.
- One filter is not enough, need to track many n-grams
- They are called 1D because we slide the window only in one direction



cat
sitting
there
or
here

0.7	**0.4**	**0.5**
0.2	**-0.1**	**0.1**
-0.5	**0.4**	**0.1**
-0.1	**0.8**	**-0.3**
-0.5	**0.3**	**0.2**

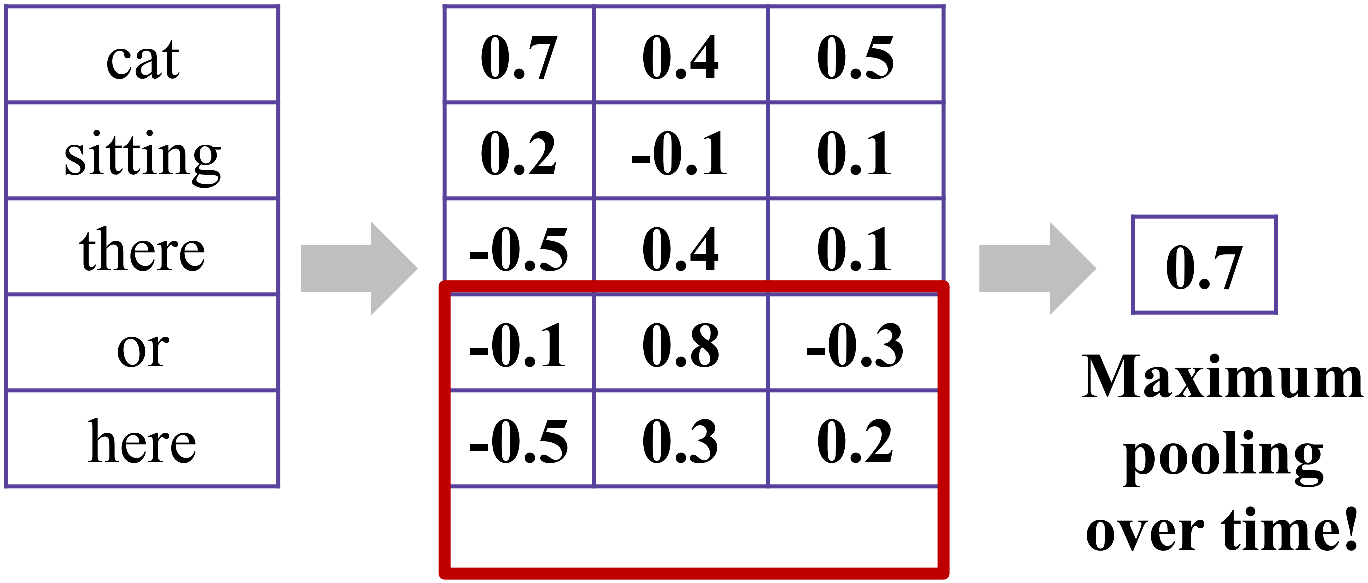
0.1
0.3
-0.2
0.7
-0.4

What to do with this vector?

can't directly use them for training as the size is depend on the input sentence length. different sentences have different feature lengths .

1D convolutions

- Can be extended to 3-grams, 4-grams, etc.
- One filter is not enough, need to track many n-grams
- They are called 1D because we slide the window only in one direction



cat
sitting
there
or
here

0.7	0.4	0.5
0.2	-0.1	0.1
-0.5	0.4	0.1
-0.1	0.8	-0.3
-0.5	0.3	0.2

0.7

Maximum pooling over time!

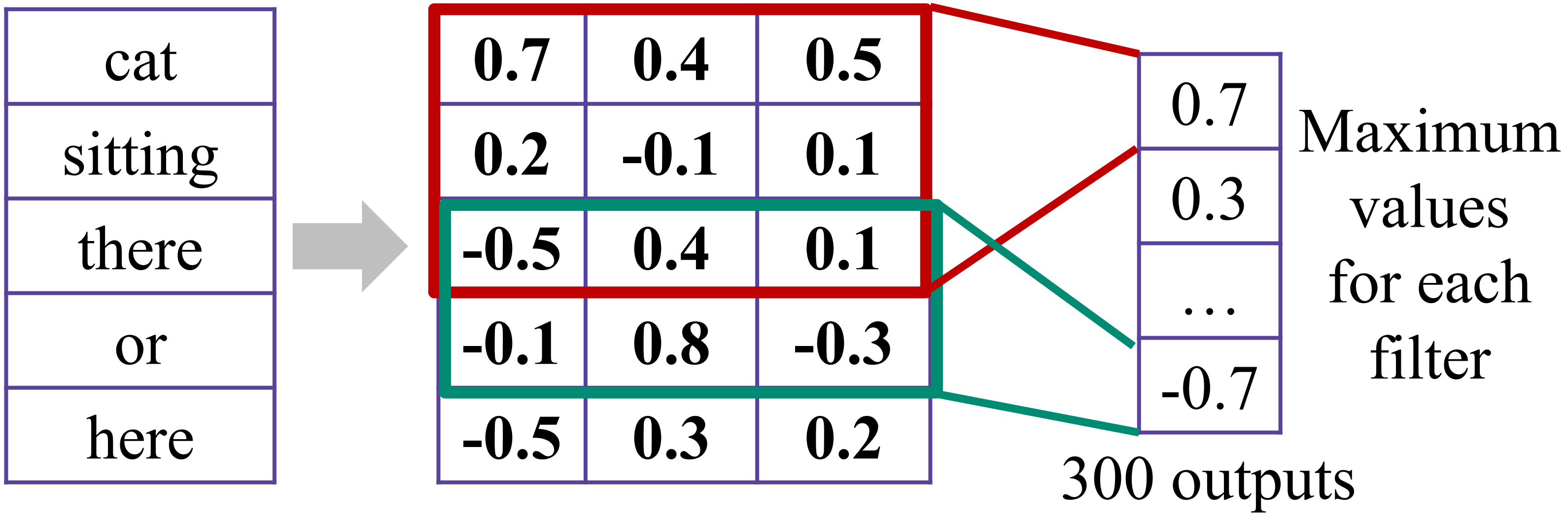
Let’s train many filters

Final architecture

- 3,4,5-gram windows with 100 filters each
- MLP on top of these 300 features

Quality comparison on customer reviews (CR)

- Naïve Bayes on top of 1,2-grams – 86.3% accuracy
- 1D convolutions with MLP – 89.6% (+3.8%) accuracy



cat
sitting
there
or
here

0.7	0.4	0.5
0.2	-0.1	0.1
-0.5	0.4	0.1
-0.1	0.8	-0.3
-0.5	0.3	0.2

0.7
0.3
…
-0.7

Maximum values for each filter

300 outputs

http://www.jmlr.org/papers/volume12/collobert11a/collobert11a.pdf https://arxiv.org/pdf/1408.5882.pdf

Summary

- You can just average pre-trained word2vec vectors for your text
- You can do better with 1D convolutions that learn more complex features
- In the next video we’ll continue to apply convolutions to text