CMPE 258, Deep Learning

Sequence learning & NLP

April 26, 2018

DMH 149A

Taehee Jeong
Ph.D., Data Scientist

SJSU SAN JOSÉ STATE UNIVERSITY

Group Project schedule

Presentation date : 5/8, 5/10

Report (including code) due date : 5/6

Number of members : 1 to 4

Content: DNN, CNN, RNN related

Platform : Pandas, Numpy, tensorflow, keras (please discuss with me for others)

Grading policy:

- Content : 40 pts
  ; Creativity in data collection, Neural network architecture / algorithm, application (same quality as a conference paper)
- Presentation : 20 pts
- Report : 20 pts
- Code : 20 pts

Applications of Image analysis using Deep Learning

- Medical Image Analysis
  : MRI, CT, Ultrasound, X-ray
- Self-driving car / drone
- Home security
- Care for the Disable
- Cloth searching

Sequential data

Text, Video, and Audio

Time series: finance, industry, medicine…

Coursera: Introduction to Deep Learning, National Research University Higher School of Economics

Applications of Recurrent Neural Networks

Speech recognition		→	"The quick brown fox jumped over the lazy dog."
Music generation	∅	→	
Sentiment classification	"There is nothing to like in this movie."	→	★☆☆☆☆
DNA sequence analysis	AGCCCCTGTGAGGAACTAG	→	AGCCCCTGTGAGGAACTAG
Machine translation	Voulez-vous chanter avec moi?	→	Do you want to sing with me?
Video activity recognition		→	Running
Name entity recognition	Yesterday, Harry Potter met Hermione Granger.	→	Yesterday, Harry Potter met Hermione Granger.

Coursera: Deep learning Specialization, Andrew Ng

Speech recognition

Amazon Echo
(Alexa)

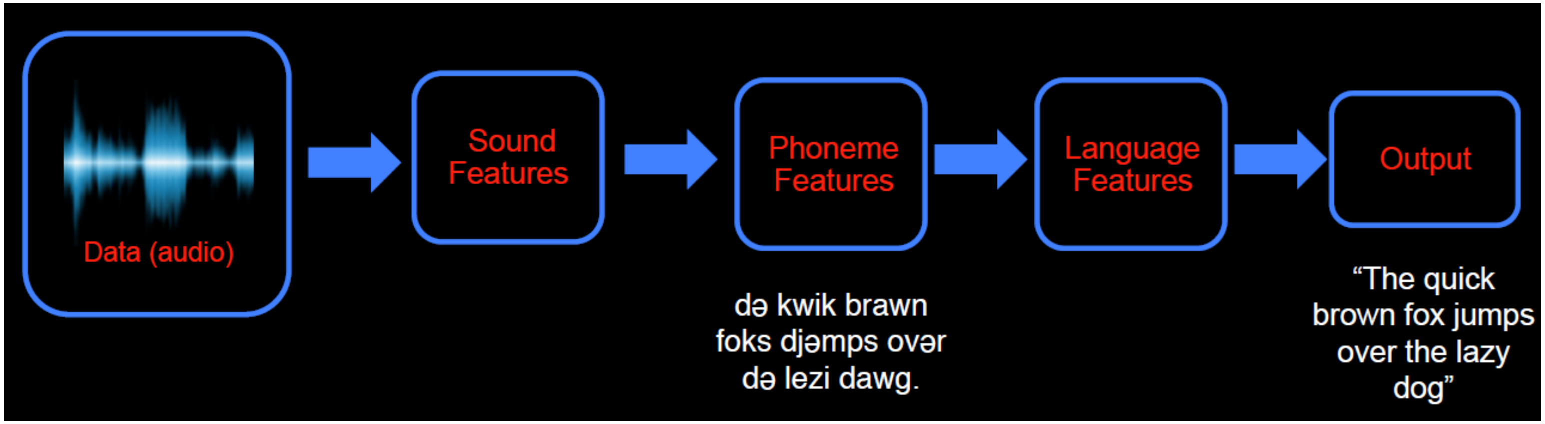
Baidu DuerOS
(xiaodunihao)

Apple Siri
(Hey Siri)

Google Home
(Okay Google)

Coursera: Deep learning Specialization, Andrew Ng

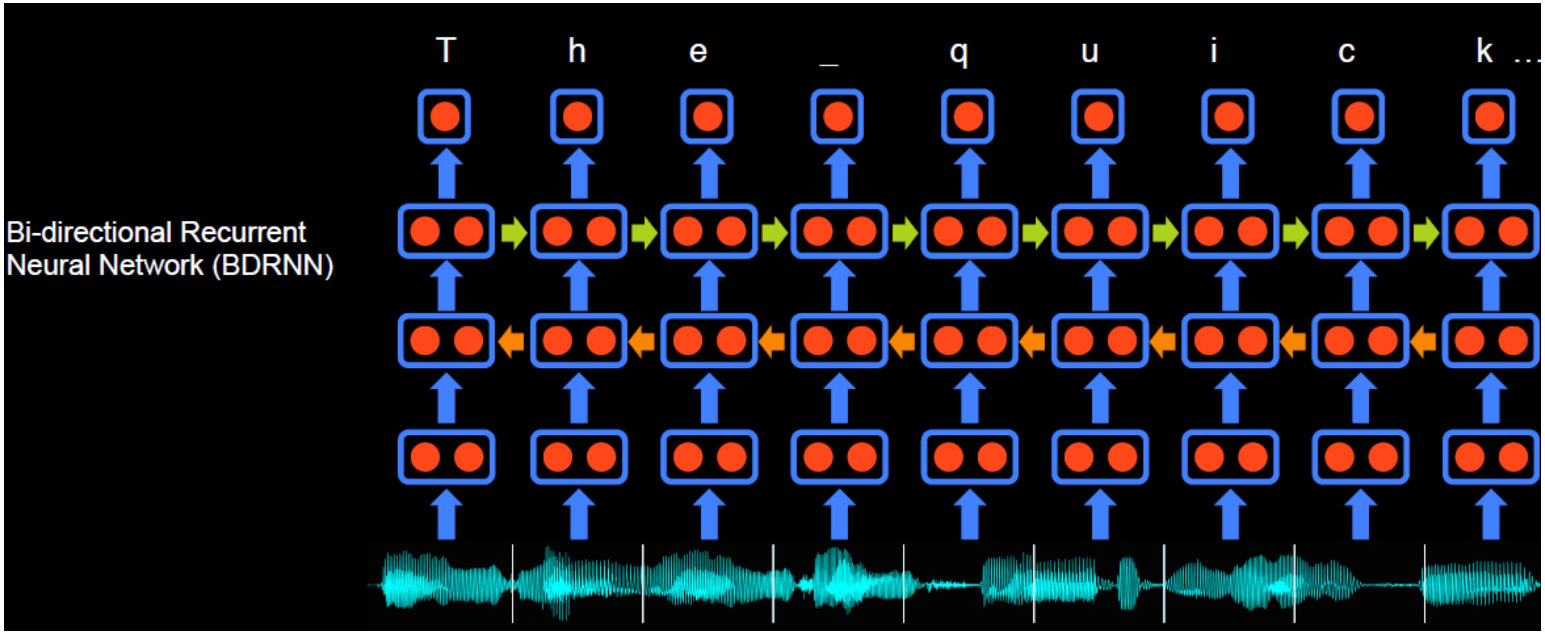
Speech recognition


Data (audio)
Sound Features
Phoneme Features
Language Features
Output
də kwik brawn foks djəmps ovər də lezi dawg.
"The quick brown fox jumps over the lazy dog"

GTC 2015, Andrew Ng

Deep speech

Baidu research



GTC 2015, Andrew Ng

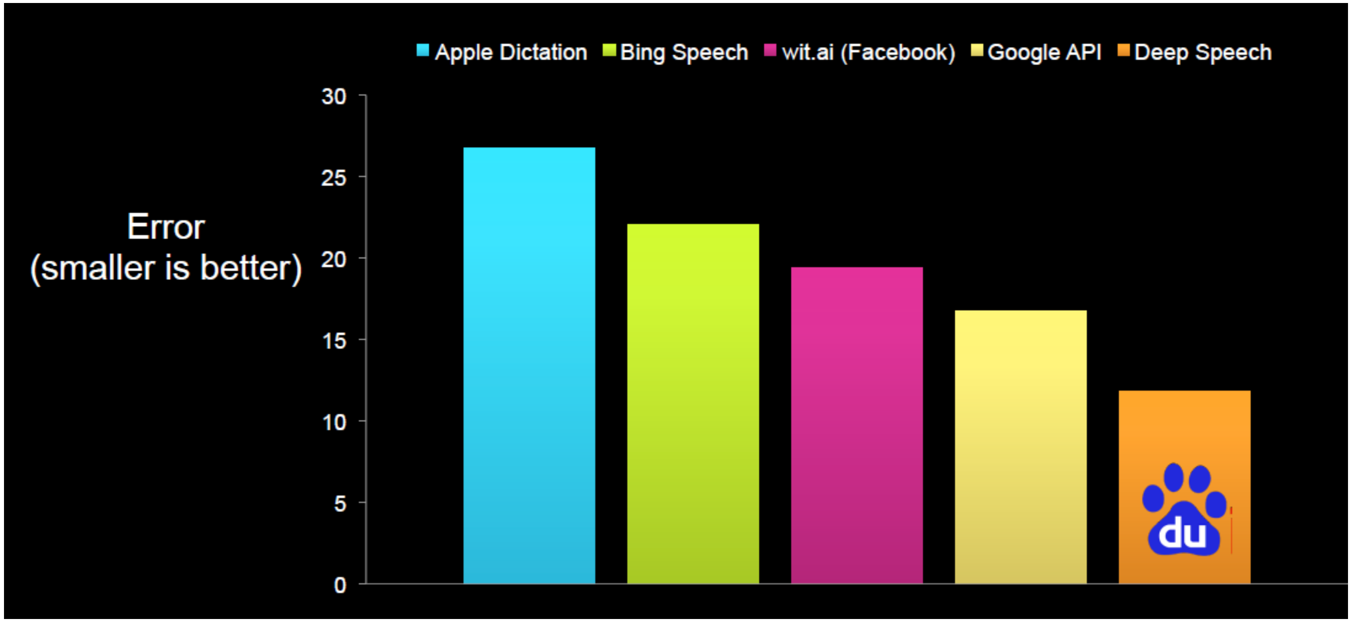
Applications of Speech recognition

- Car interfaces
- Home appliances
- Wearables
- Care for the Disable

GTC 2015, Andrew Ng

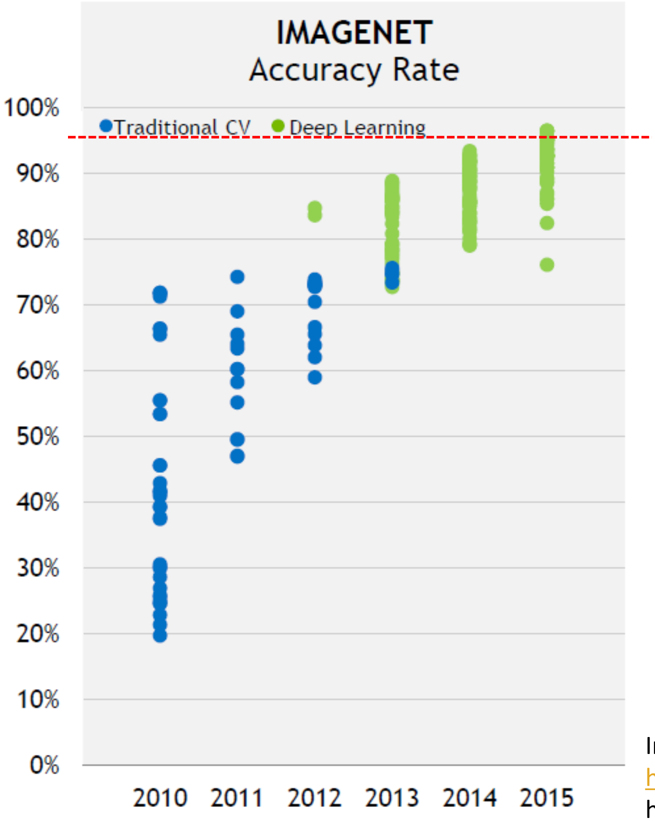
Speech recognition errors

There are room to improve compared to image recognition.



GTC 2015, Andrew Ng

Image classification

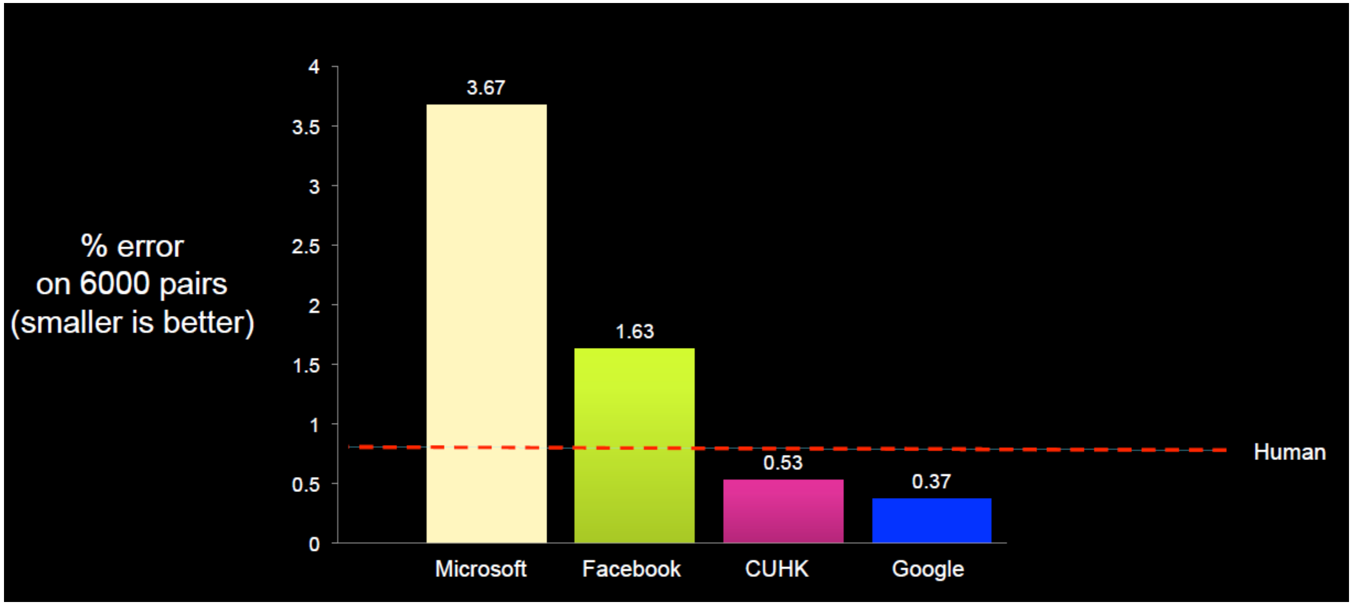


Top 5 error rate for image classification in ILSVRC ImageNet challenge
From 26% to 3% in six years

Human accuracy : 95 ~ 97 %

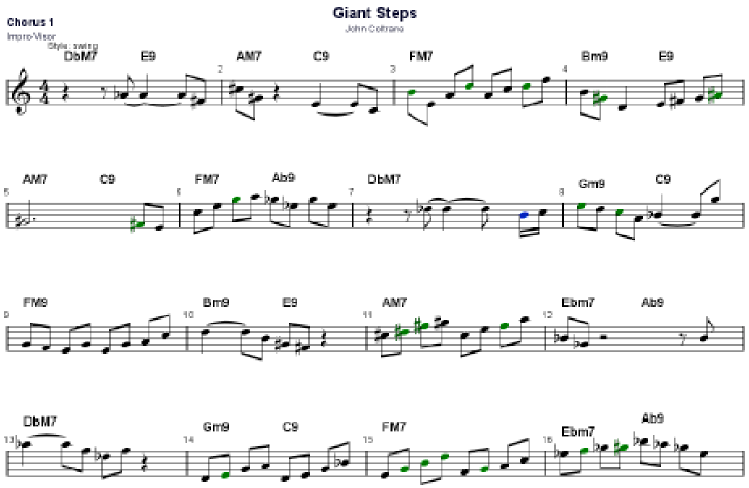
Image-Net Large-Scale Visual Recognition Challenge (ILSVRC)
http://image-net.org/challenges/LSVRC/
https://www.slideshare.net/NVIDIA/nvidia-ces-2016-press-conference

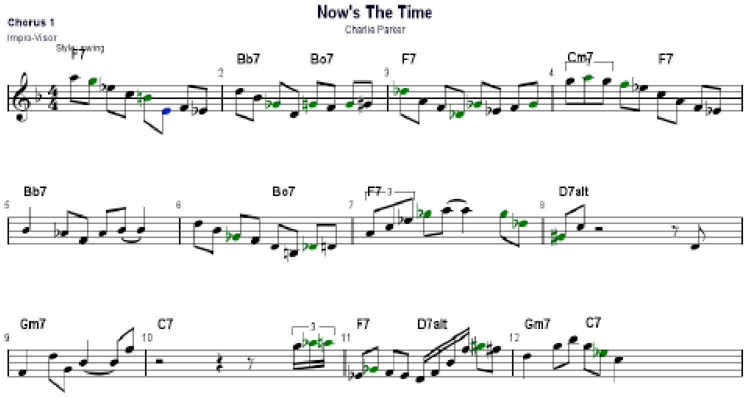
Face recognition errors



GTC 2015, Andrew Ng

Music generation by RNN





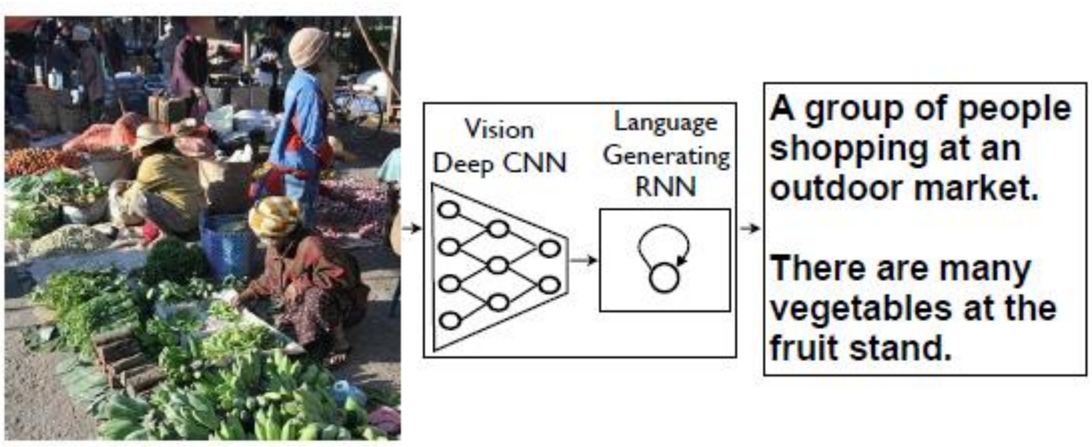
LEARNING JAZZ GRAMMARS, SMC09, Jon Gillick et al.

Text generation using RNN

They had no choice but the most recent univerbeen fairly uncomfortable and dangerous as ever. As long as he dived experience that it was not uncertain that even Harry had taken in black tail as the train roared and was thin, but Harry, Ron, and Hermione, at the fact that he was in complete disarraying the rest of the class holding him, he should have been able to prove them.

https://chunml.github.io/ChunML.github.io/project/Creating-Text-Generator-Using-Recurrent-Neural-Network/

A Neural Image Caption Generator

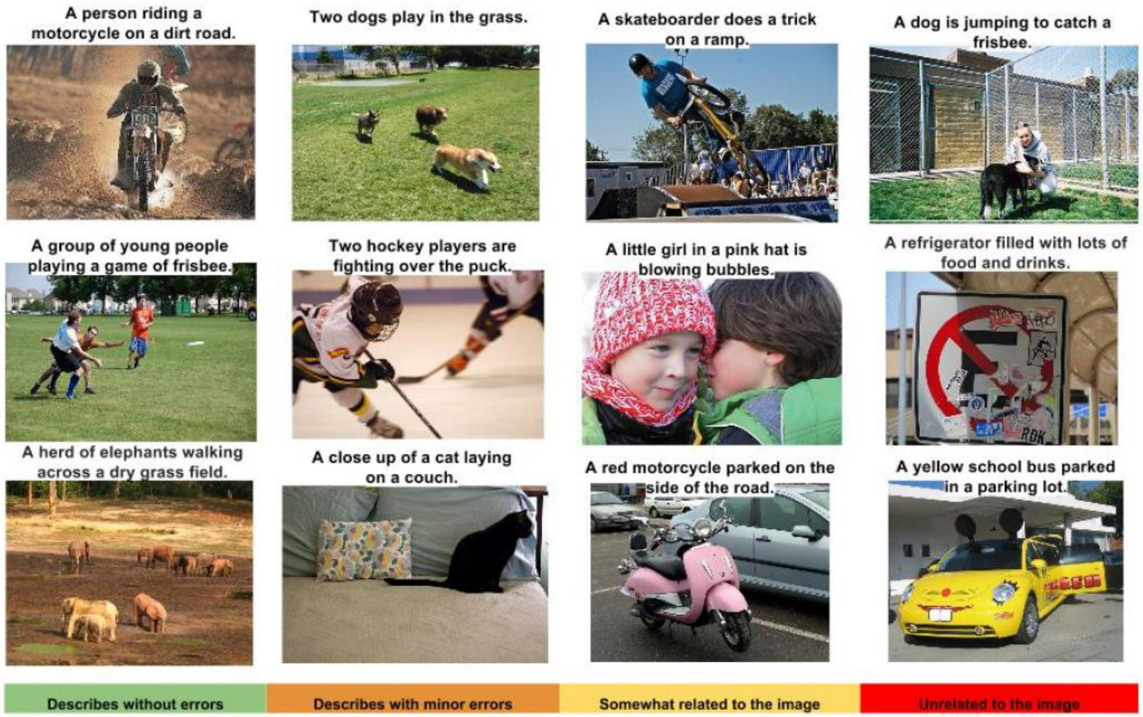


A group of people shopping at an outdoor market.

There are many vegetables at the fruit stand.

Show and Tell: A Neural Image Caption Generator, Oriol Vinyals et al., 2015,

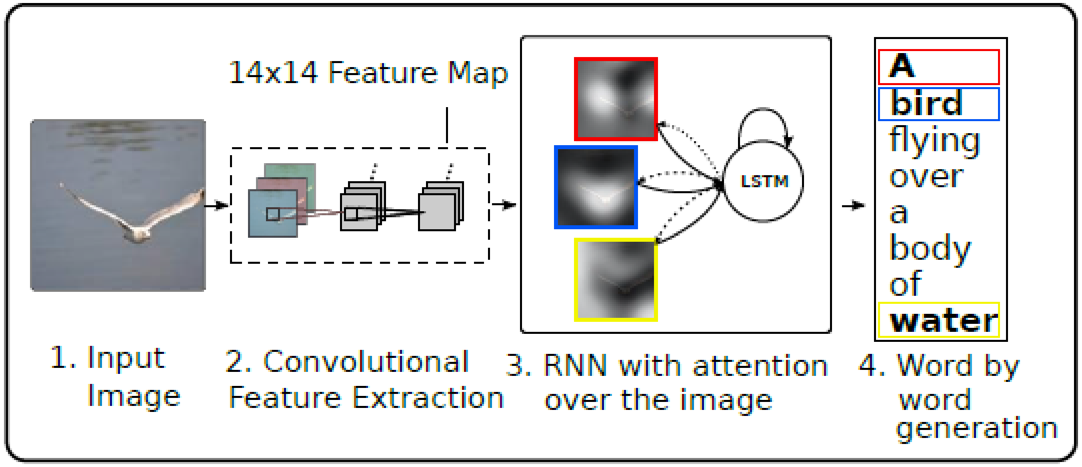
A Neural Image Caption Generator


A person riding a motorcycle on a dirt road.
Two dogs play in the grass.
A skateboarder does a trick on a ramp.
A dog is jumping to catch a frisbee.
A group of young people playing a game of frisbee.
Two hockey players are fighting over the puck.
A little girl in a pink hat is blowing bubbles.
A refrigerator filled with lots of food and drinks.
A herd of elephants walking across a dry grass field.
A close up of a cat laying on a couch.
A red motorcycle parked on the side of the road.
A yellow school bus parked in a parking lot.
Describes without errors
Describes with minor errors
Somewhat related to the image
Unrelated to the image

Show and Tell: A Neural Image Caption Generator, Oriol Vinyals et al., 2015,

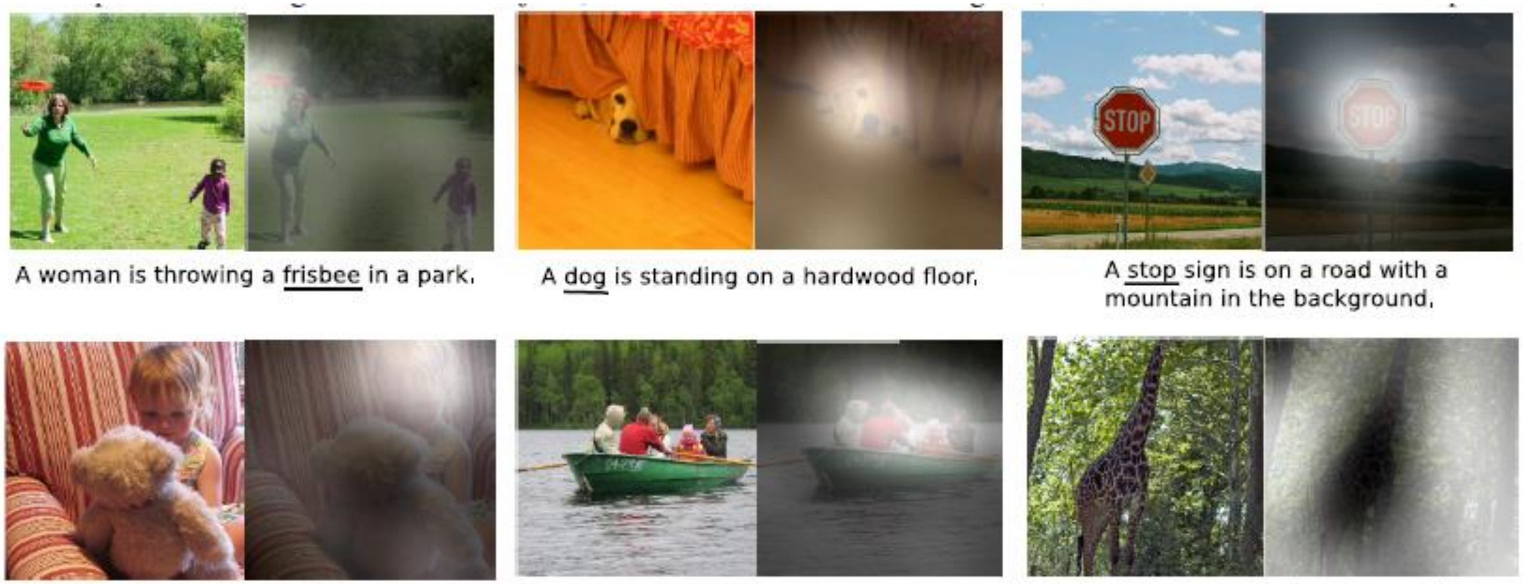
Neural Image Caption Generation with Visual Attention

Visualized attentional maps


14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, Kelvin Xu et al., 2016

Neural Image Caption Generation with Visual Attention



A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, Kelvin Xu et al., 2016

Natural Language Process (NLP)

- Speech recognition
- Machine translation
- Chatbots (question answering)
- Sentiment classification
- Name entity recognition

Approach for NLP

- Machine Learning
- Convolutional Neural Network
- Recurrent Neural Network

Sentiment analysis

Text is a sequence of words

Word is a sequence of characters

How to separate words from a sentence?

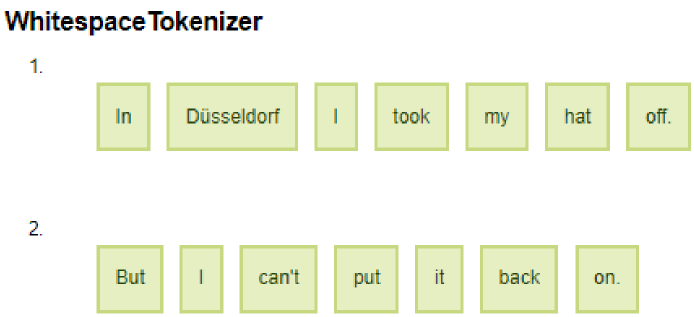
Tokenization

Tokenization is a process that splits an input sequence into tokens.

We can split token by space, punctuation, a set of rule.

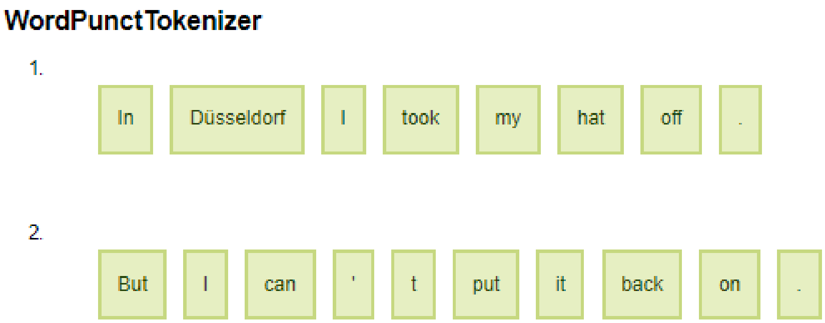
Tokenization

In Düsseldorf I took my hat off. But I can't put it back on.



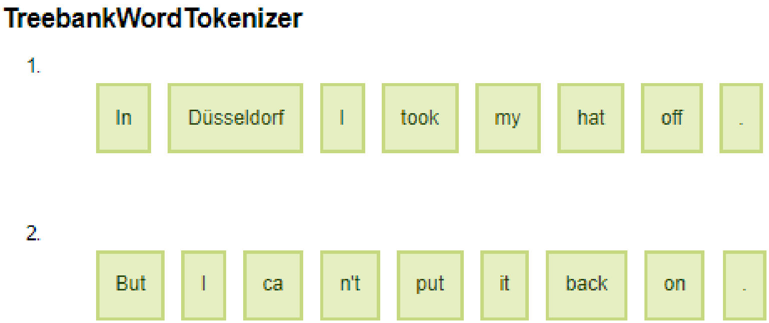
WhitespaceTokenizer

1.	In	Düsseldorf	I	took	my	hat	off.
2.	But	I	can't	put	it	back	on.



WordPunctTokenizer

1.	In	Düsseldorf	I	took	my	hat	off	.		
2.	But	I	can	'	t	put	it	back	on	.

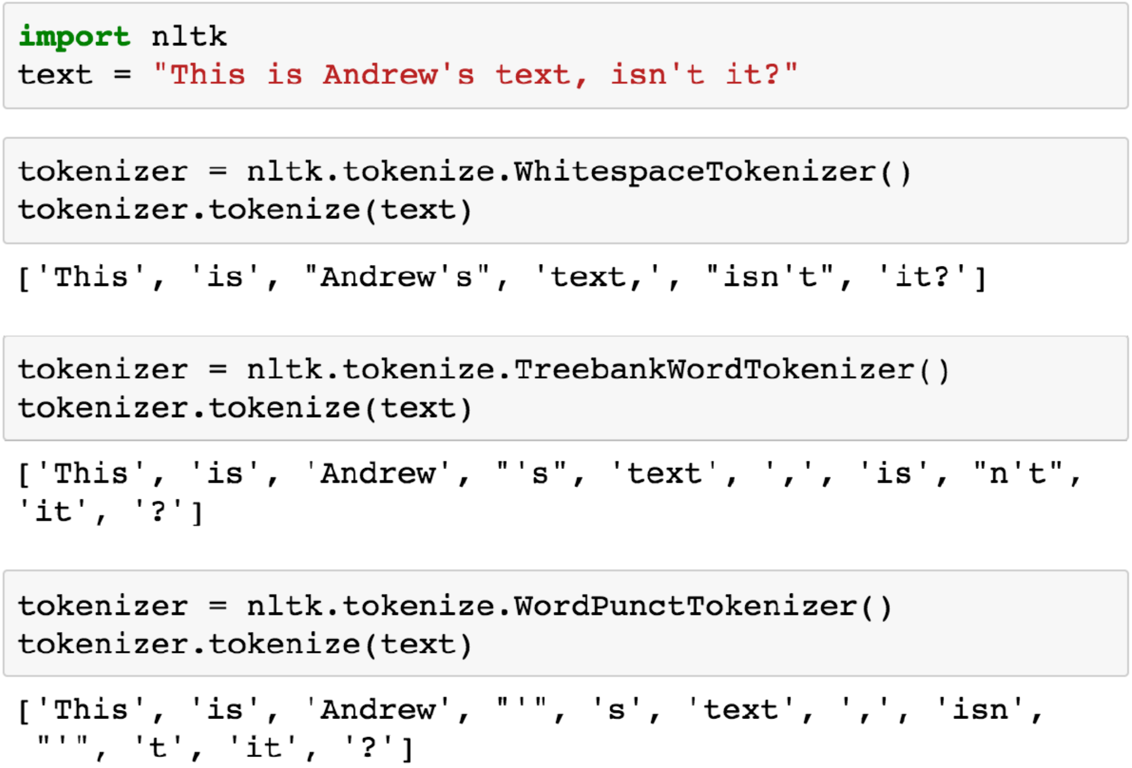


TreebankWordTokenizer

1.	In	Düsseldorf	I	took	my	hat	off	.	
2.	But	I	ca	n't	put	it	back	on	.

http://text-processing.com/demo/tokenize/

Python tokenization example



```
import nltk
text = "This is Andrew's text, isn't it?"
```

```
tokenizer = nltk.tokenize.WhitespaceTokenizer()
tokenizer.tokenize(text)
```

```
['This', 'is', "Andrew's", 'text,', "isn't", 'it?']
```

```
tokenizer = nltk.tokenize.TreebankWordTokenizer()
tokenizer.tokenize(text)
```

```
['This', 'is', 'Andrew', "'s", 'text', ',', 'is', "n't",
'it', '?']
```

```
tokenizer = nltk.tokenize.WordPunctTokenizer()
tokenizer.tokenize(text)
```

```
['This', 'is', 'Andrew', "'", 's', 'text', ',', 'isn',
 "'", 't', 'it', '?']
```

Coursera: Natural Language Processing, National Research University Higher School of Economics

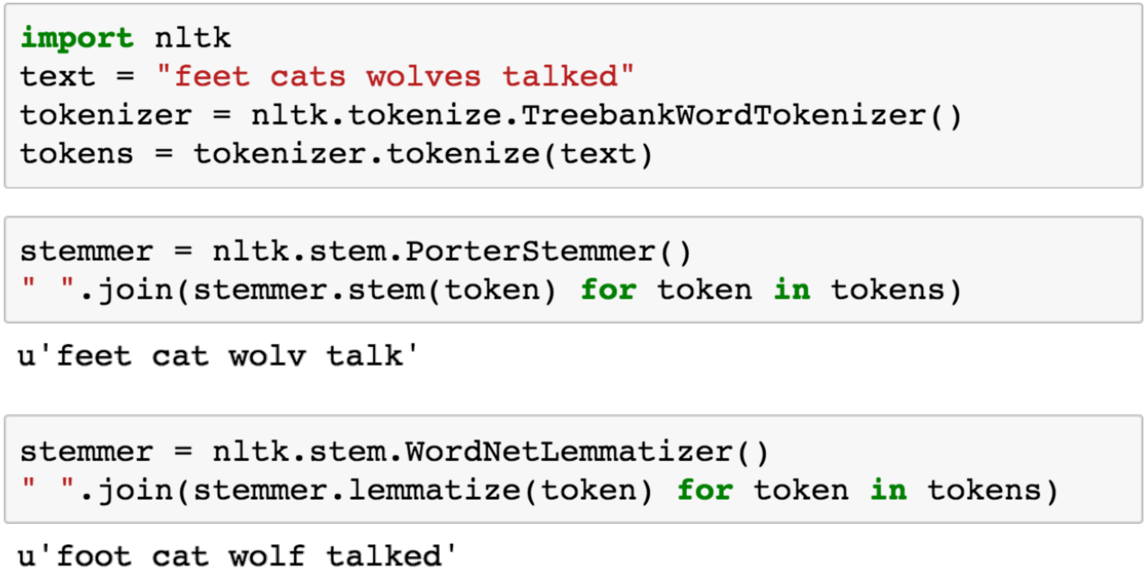
Token normalization

Same token for different forms of words

- Examples
  - wolf, wolves → wolf
  - talk, talks → talk
- Stemming
  - removes and replaces suffixes to get to the root form of a word, which is called as stem.
- Lemmatization
  - returns the base or dictionary form of a word, which is known as lemma.

Coursera: Natural Language Processing, National Research University Higher School of Economics

Python stemming example



```python
import nltk
text = "feet cats wolves talked"
tokenizer = nltk.tokenize.TreebankWordTokenizer()
tokens = tokenizer.tokenize(text)
```

```python
stemmer = nltk.stem.PorterStemmer()
" ".join(stemmer.stem(token) for token in tokens)
```

```
u'feet cat wolv talk'
```

```python
stemmer = nltk.stem.WordNetLemmatizer()
" ".join(stemmer.lemmatize(token) for token in tokens)
```

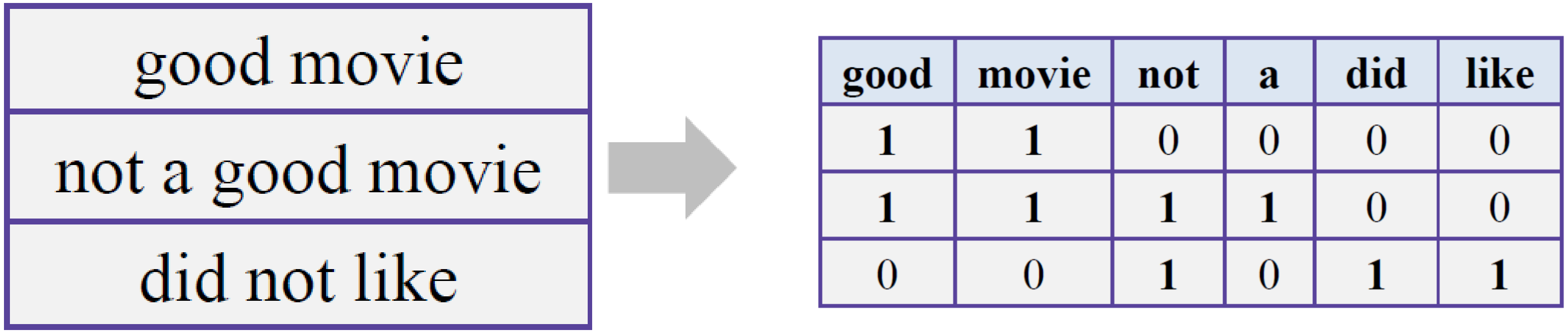
```
u'foot cat wolf talked'
```

Coursera: Natural Language Processing, National Research University Higher School of Economics

Transforming tokens into features

Bag of words (BOW)

For each token, we have a feature column, which is called text vectorization.



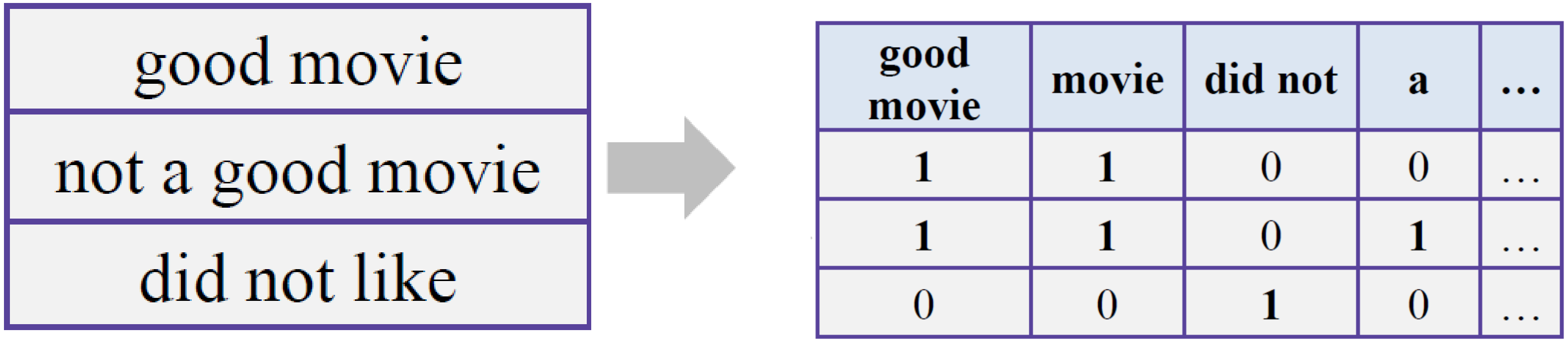
good movie
not a good movie
did not like

good	movie	not	a	did	like
1	1	0	0	0	0
1	1	1	1	0	0
0	0	1	0	1	1

Coursera: Natural Language Processing, National Research University Higher School of Economics

Preserve some ordering

N-grams: Token pairs, triplets, etc.



good movie
not a good movie
did not like

good movie	movie	did not	a	…
1	**1**	0	0	…
1	**1**	0	**1**	…
0	0	**1**	0	…

Coursera: Natural Language Processing, National Research University Higher School of Economics

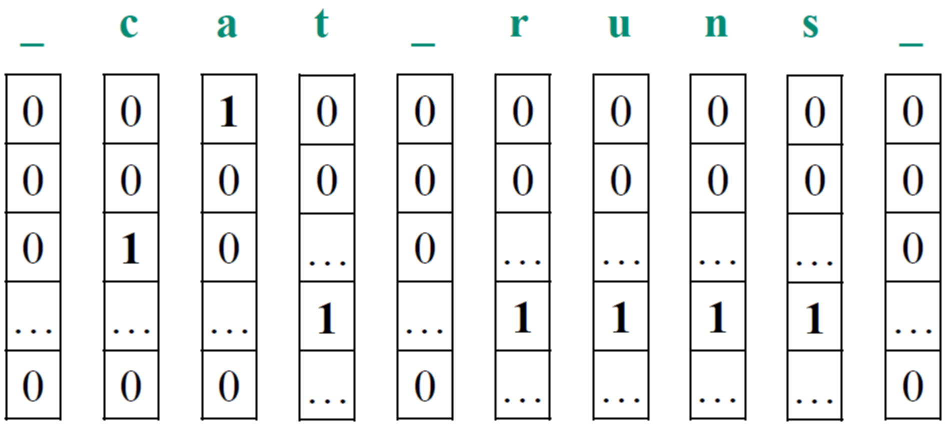
Remove some n-grams

- High frequency n-grams
  - Articles, prepositions, etc. (example: and, a, the)
  - They are called stop-words. They do not help to discriminate texts.
- Low frequency n-grams
  - Typos, rare words

Coursera: Natural Language Processing, National Research University Higher School of Economics

Text as a sequence of characters

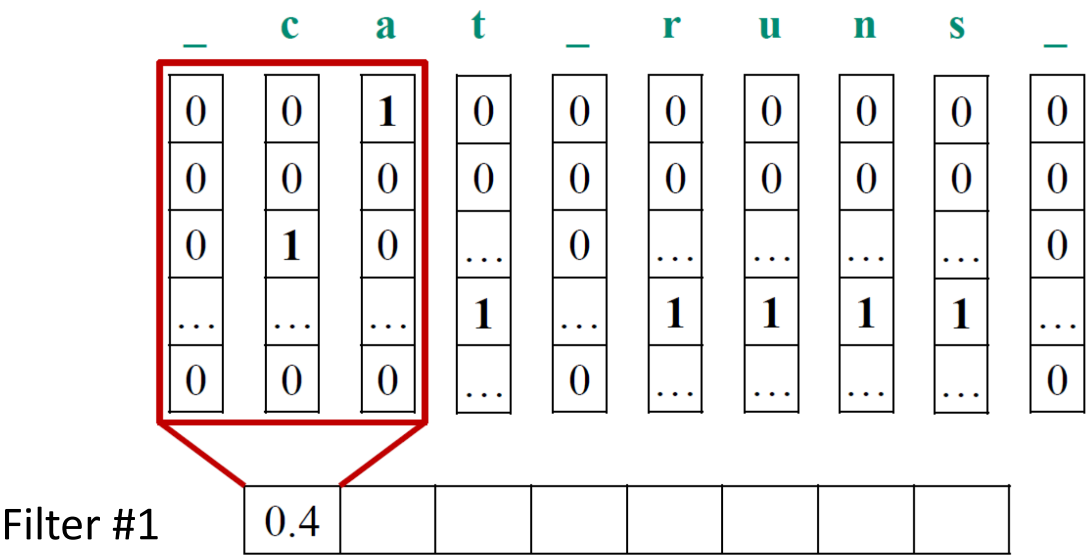
One-hot encoding characters, length ~ 70



_	c	a	t	_	r	u	n	s	_
0	0	**1**	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	**1**	0	...	0	...	...	...	...	0
...	...	...	**1**	...	**1**	**1**	**1**	**1**	...
0	0	0	...	0	...	...	...	...	0

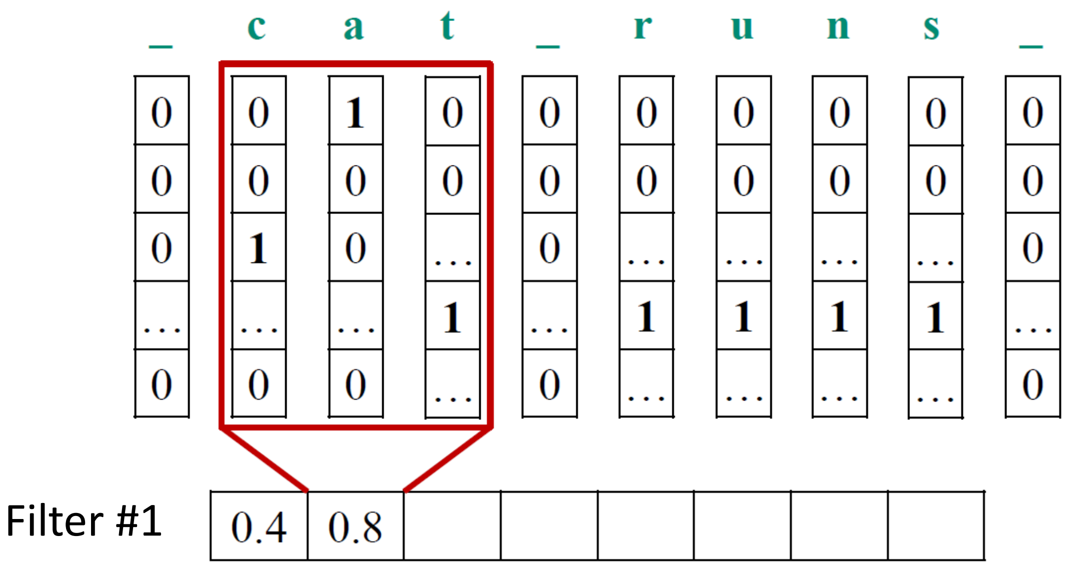
Coursera: Natural Language Processing, National Research University Higher School of Economics

1D convolutions on characters



Coursera: Natural Language Processing, National Research University Higher School of Economics

1D convolutions on characters

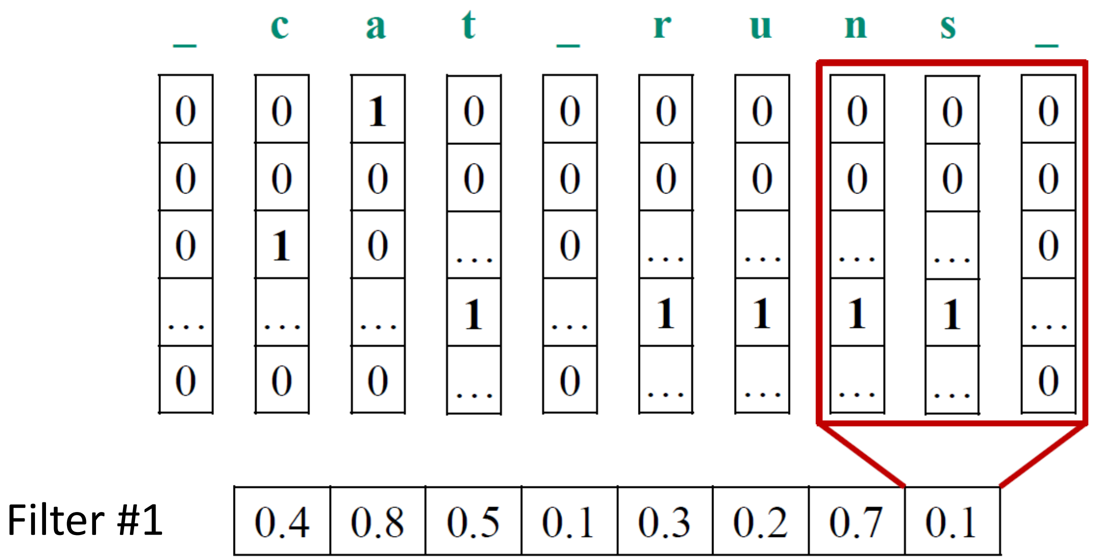


_	c	a	t	_	r	u	n	s	_
0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	1	0	…	0	…	…	…	…	0
…	…	…	1	…	1	1	1	1	…
0	0	0	…	0	…	…	…	…	0

Filter #1	0.4	0.8						

Coursera: Natural Language Processing, National Research University Higher School of Economics

1D convolutions on characters

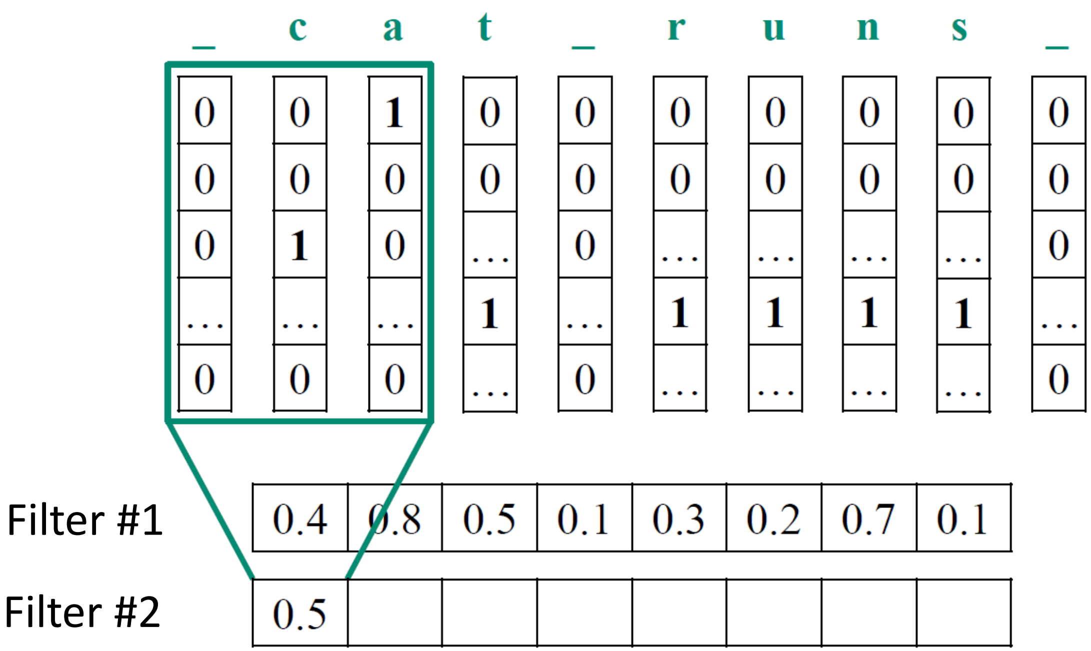


_	c	a	t	_	r	u	n	s	_
0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	1	0	...	0	...	...	...	...	0
...	...	...	1	...	1	1	1	1	...
0	0	0	...	0	...	...	...	...	0

Filter #1	0.4	0.8	0.5	0.1	0.3	0.2	0.7	0.1

Coursera: Natural Language Processing, National Research University Higher School of Economics

1D convolutions on characters

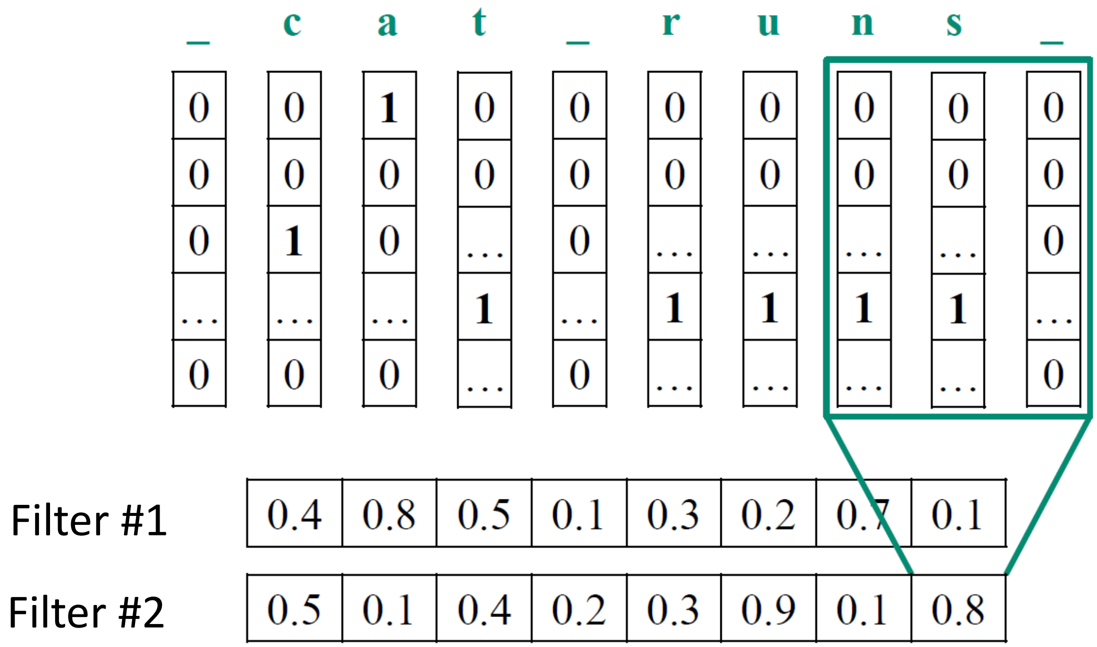


_	c	a	t	_	r	u	n	s	_
0	0	**1**	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	**1**	0	…	0	…	…	…	…	0
…	…	…	**1**	…	**1**	**1**	**1**	**1**	…
0	0	0	…	0	…	…	…	…	0

Filter #1	0.4	0.8	0.5	0.1	0.3	0.2	0.7	0.1
Filter #2	0.5							

Coursera: Natural Language Processing, National Research University Higher School of Economics

1D convolutions on characters



_	c	a	t	_	r	u	n	s	_
0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	1	0	...	0	...	...	...	...	0
...	...	...	1	...	1	1	1	1	...
0	0	0	...	0	...	...	...	...	0

Filter #1	0.4	0.8	0.5	0.1	0.3	0.2	0.7	0.1
Filter #2	0.5	0.1	0.4	0.2	0.3	0.9	0.1	0.8

Coursera: Natural Language Processing, National Research University Higher School of Economics

1D convolutions on characters

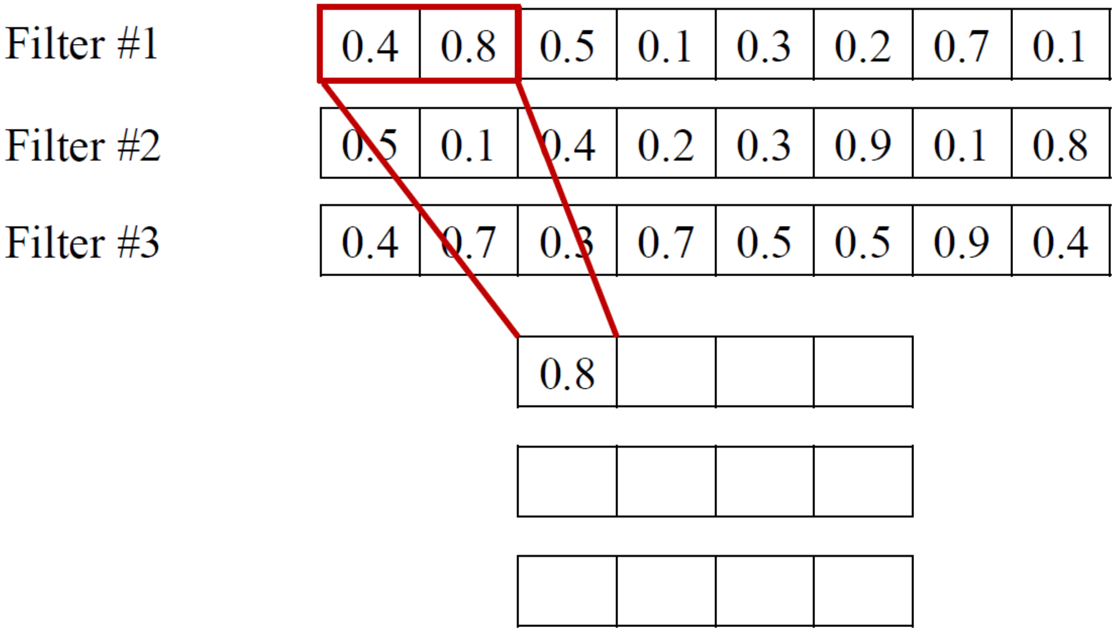
_	c	a	t	_	r	u	n	s	_
0	0	**1**	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	**1**	0	…	0	…	…	…	…	0
…	…	…	**1**	…	**1**	**1**	**1**	**1**	…
0	0	0	…	0	…	…	…	…	0

Filter #1	0.4	0.8	0.5	0.1	0.3	0.2	0.7	0.1
Filter #2	0.5	0.1	0.4	0.2	0.3	0.9	0.1	0.8
Filter #3	0.4	0.7	0.3	0.7	0.5	0.5	0.9	0.4

Coursera: Natural Language Processing, National Research University Higher School of Economics

Max pooling

Provides position invariance for character n-gram



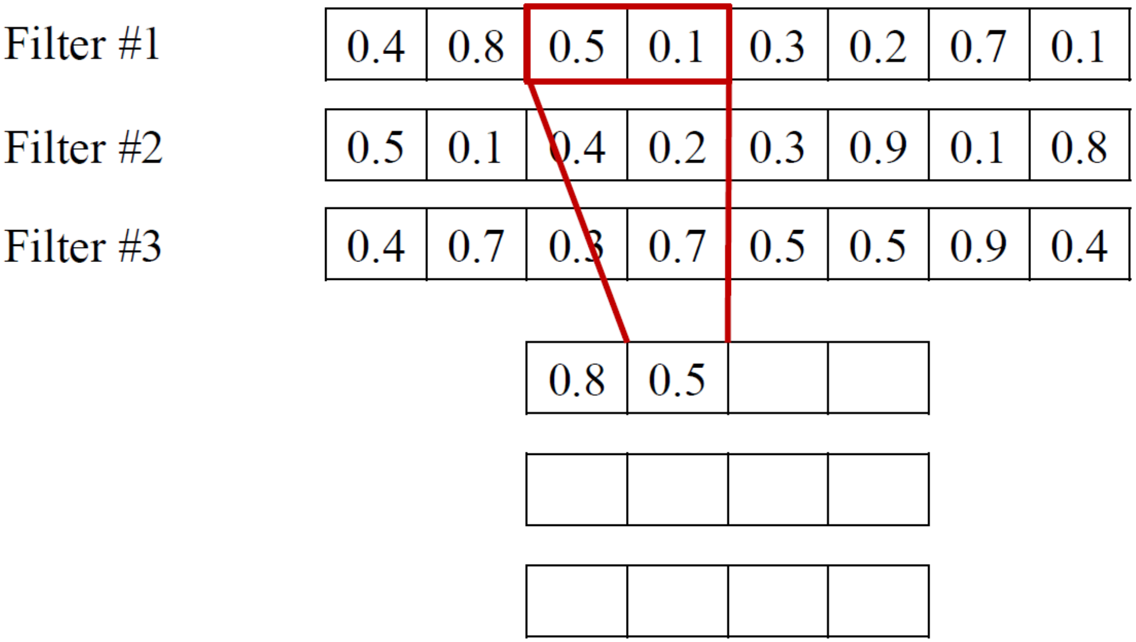
Filter #1	0.4	0.8	0.5	0.1	0.3	0.2	0.7	0.1
Filter #2	0.5	0.1	0.4	0.2	0.3	0.9	0.1	0.8
Filter #3	0.4	0.7	0.3	0.7	0.5	0.5	0.9	0.4

0.8			

Coursera: Natural Language Processing, National Research University Higher School of Economics

Max pooling

Provides position invariance for character n-gram



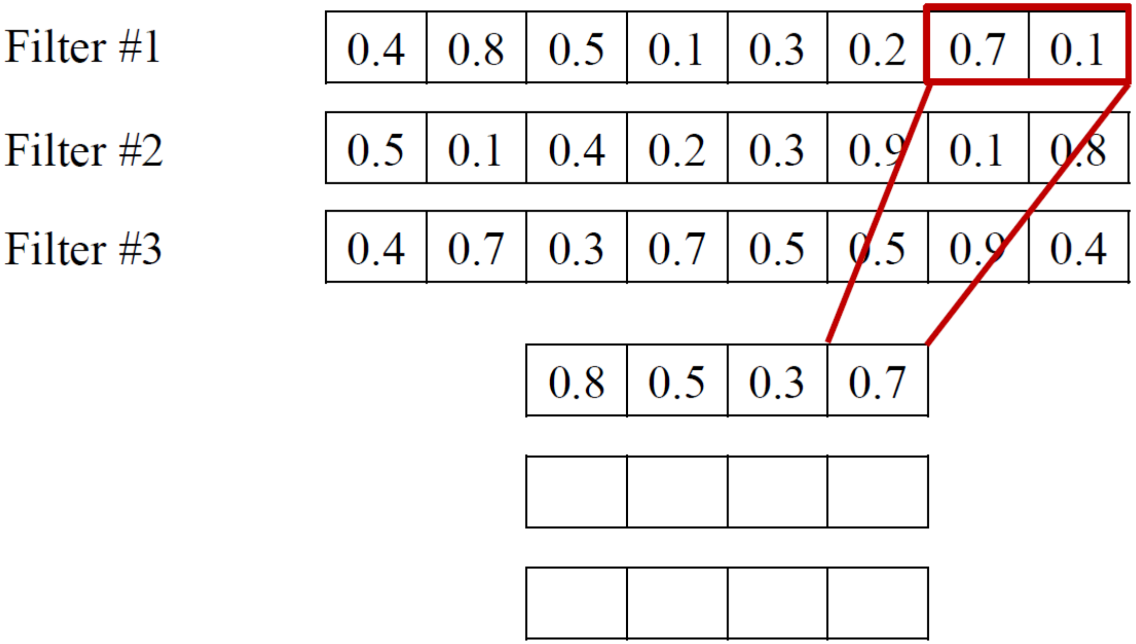
Filter #1	0.4	0.8	0.5	0.1	0.3	0.2	0.7	0.1
Filter #2	0.5	0.1	0.4	0.2	0.3	0.9	0.1	0.8
Filter #3	0.4	0.7	0.3	0.7	0.5	0.5	0.9	0.4

0.8	0.5		

Coursera: Natural Language Processing, National Research University Higher School of Economics

Max pooling

Provides position invariance for character n-gram



Filter #1	0.4	0.8	0.5	0.1	0.3	0.2	0.7	0.1
Filter #2	0.5	0.1	0.4	0.2	0.3	0.9	0.1	0.8
Filter #3	0.4	0.7	0.3	0.7	0.5	0.5	0.9	0.4

0.8	0.5	0.3	0.7

Coursera: Natural Language Processing, National Research University Higher School of Economics

Max pooling

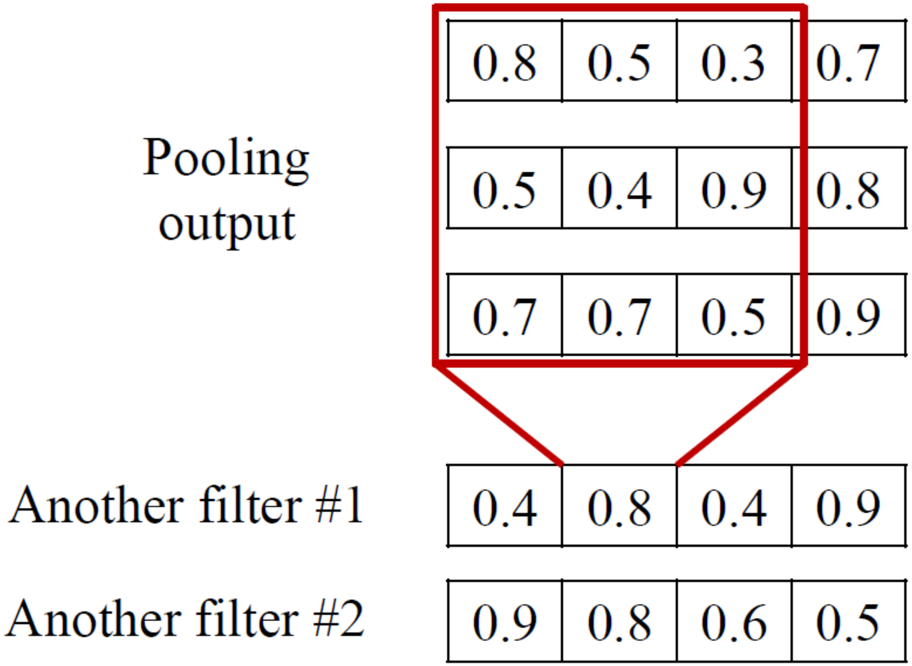
Provides position invariance for character n-gram

Filter #1	0.4	0.8	0.5	0.1	0.3	0.2	0.7	0.1
Filter #2	0.5	0.1	0.4	0.2	0.3	0.9	0.1	0.8
Filter #3	0.4	0.7	0.3	0.7	0.5	0.5	0.9	0.4

Pooling output

0.8	0.5	0.3	0.7
0.5	0.4	0.9	0.8
0.7	0.7	0.5	0.9

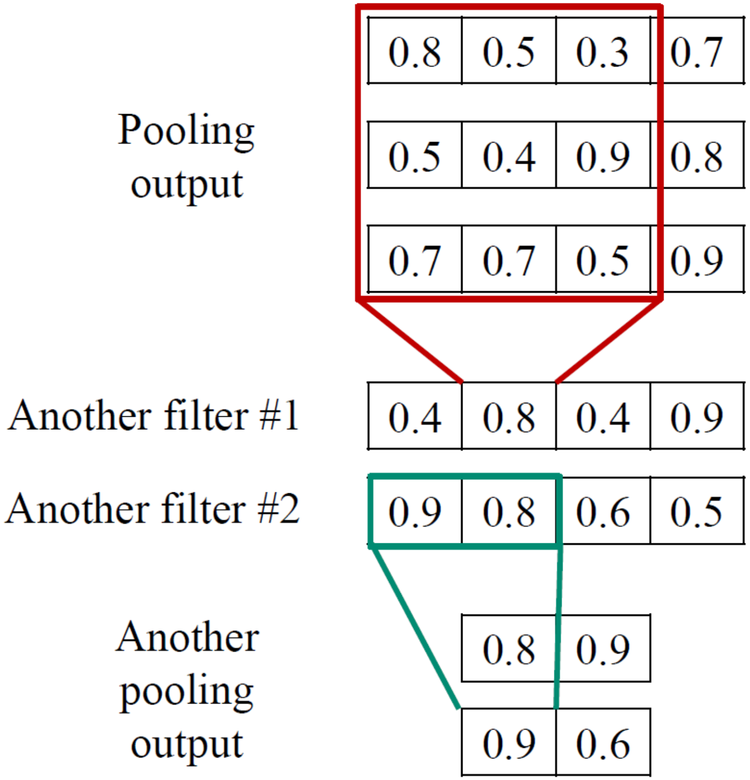
Repeat 1D convolutions + pooling



Pooling output	0.8	0.5	0.3	0.7
	0.5	0.4	0.9	0.8
	0.7	0.7	0.5	0.9
Another filter #1	0.4	0.8	0.4	0.9
Another filter #2	0.9	0.8	0.6	0.5

Coursera: Natural Language Processing, National Research University Higher School of Economics

Repeat 1D convolutions + pooling



Pooling output

0.8	0.5	0.3	0.7
0.5	0.4	0.9	0.8
0.7	0.7	0.5	0.9

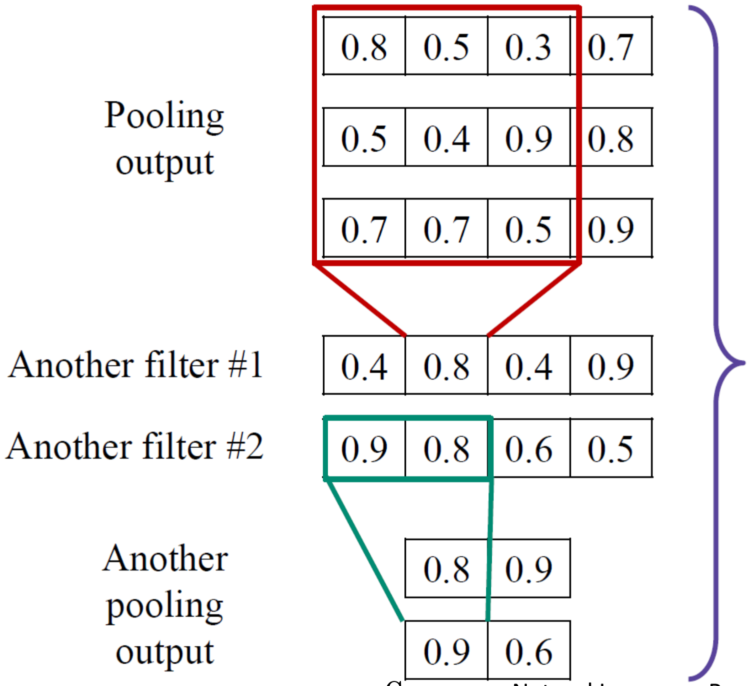
Another filter #1	0.4	0.8	0.4	0.9
Another filter #2	0.9	0.8	0.6	0.5

Another pooling output

0.8	0.9
0.9	0.6

Coursera: Natural Language Processing, National Research University Higher School of Economics

Repeat 1D convolutions + pooling



Pooling output

0.8	0.5	0.3	0.7
0.5	0.4	0.9	0.8
0.7	0.7	0.5	0.9

Another filter #1	0.4	0.8	0.4	0.9
Another filter #2	0.9	0.8	0.6	0.5

Another pooling output

0.8	0.9
0.9	0.6

Repeat many times

Coursera: Natural Language Processing, National Research University Higher School of Economics

Example of 1D convolution architecture for chacacters

- Characters of text : 1014
- Apply 1D convolution + max pooling 6 times
- Kernels widths : 7,7,3,3,3,3
- # of Filters at each step : 1024
- Output: 1024 x 34 matrix of features
- Apply multiclass classification for sentiment analysis

Coursera: Natural Language Processing, National Research University Higher School of Economics

Summary

- Sequence data and its application using RNN
- Natural Language Process
- Tokenization
- Bag of Words, Word vector
- 1D convolution with characters