

# Revision of week 3 and 4

Nhung Nguyen and Viktor Schlegel

# Week 3 - Distributional semantics

We shall know a word by the company it keeps.

Firth (1957)



# Week 3 - Count-based approach

- Term-document matrix

	As You Like It	Twelfth Night	Julius Caesar	Henry V
<b>battle</b>	1	0	7	13
<b>good</b>	114	80	62	89
<b>fool</b>	36	58	1	4
<b>wit</b>	20	15	2	3

tf-idf

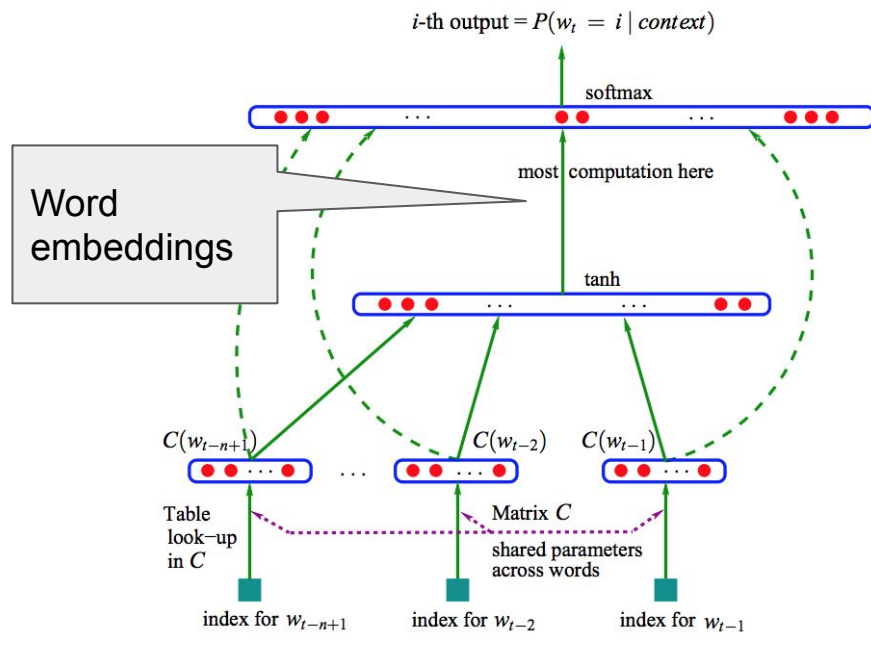
- Term-term matrix

	aardvark	computer	data	pinch	result	sugar	...
<b>apricot</b>	0	0	0	1	0	1	
<b>pineapple</b>	0	0	0	1	0	1	
<b>digital</b>	0	2	1	0	1	0	
<b>information</b>	0	1	6	0	4	0	

smoothing

# Week 3 - Prediction-based approach

- Bengio's language model



$$\Pr(w_t | w_{t-1}, \dots, w_{t-m+1}) = \text{softmax}(\mathbf{W}\mathbf{y})$$

$$\mathbf{y} = \tanh(\mathbf{V}\mathbf{x})$$

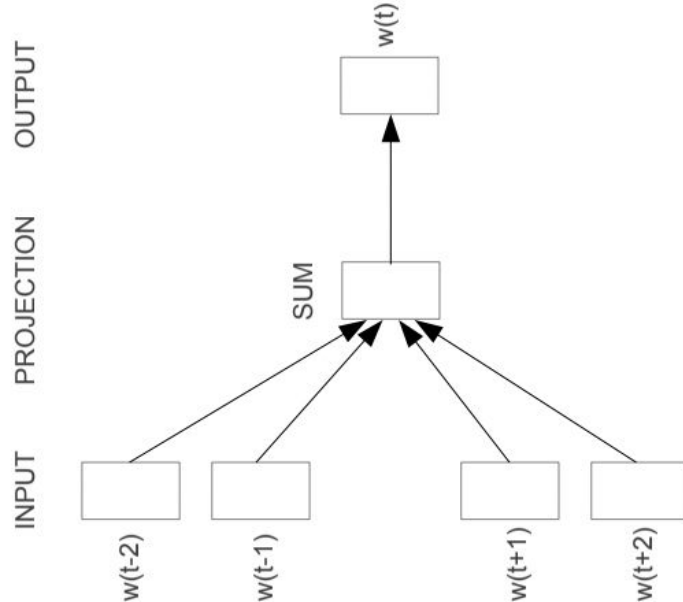
$$\mathbf{x} = \text{concat}(\mathbf{w}_{t-1}, \dots, \mathbf{w}_{t-m+1})$$

Each *contextual* word  $w_{t-j}$  is represented a column of matrix  $\mathbf{C}$

# Week 3 - Prediction-based approach (Cont.)

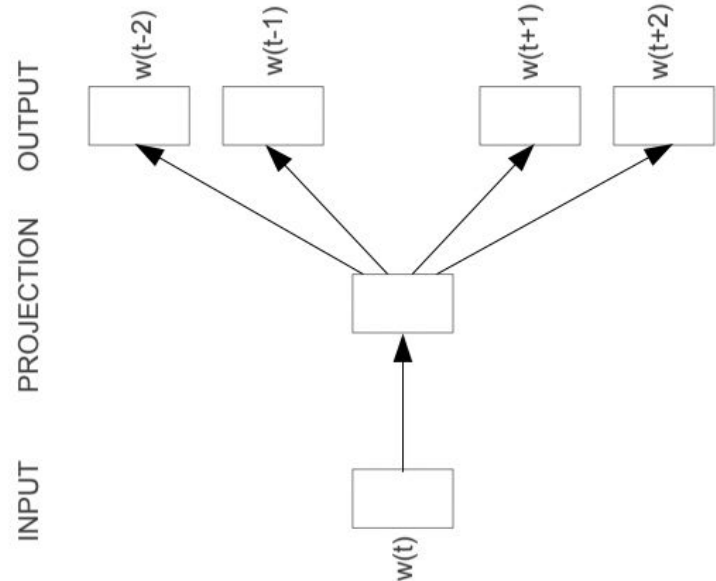
- Mikolov's CBOW

$$\Pr(w_t | w_{t-1}, \dots, w_{t-m+1}) = \text{softmax}(\mathbf{W}\mathbf{y})$$



- Mikolov's Skip-gram

$$\Pr(\cdot | w_t) = \text{softmax}(\mathbf{W}\mathbf{y})$$



# Week 3 - Word Sense Disambiguation

- WordNet: a database of lexical relations

- A word has different senses

**mouse**<sup>1</sup> : .... a *mouse* controlling a computer system in 1968.

**mouse**<sup>2</sup> : .... a quiet animal like a *mouse*

**bank**<sup>1</sup> : ...a *bank* can hold the investments in a custodial account ...

**bank**<sup>2</sup> : ...as agriculture burgeons on the east *bank*, the river ...

- Disambiguating word sense using Lesk's algorithm and supervised learning

# Week 4 - Sequence modelling

## Sequence classification

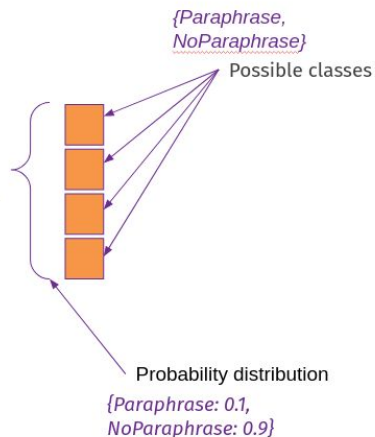
S1: I like trains.

S2: The train is arriving on time.

Input text

The Reds won the AFC last for the 10th straight year. The Patriots trailed 24-10 at the end of the third quarter. They scored on a 40-yard field goal with 4:01 left in the game to pull within 26-10. Then, with 30 seconds remaining, Steve Lavelle scored on an 8-yard run and the Patriots added a two-point conversion to go ahead 27-26. The game ended on a Scottie Hunter interception. Quarterback Tom Brady threw an interception in the first half with a broken calf.

Classifier



Applicable NLP tasks:

- Sentiment analysis
- textual entailment
- paraphrasing
- question type classification

## Span extraction

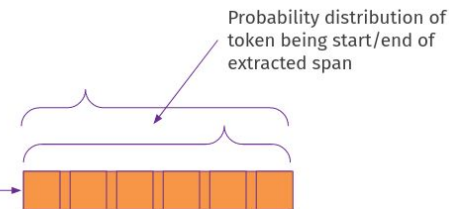
P: I like trains.

Q: Who likes trains?

Input text

The Reds won the AFC last for the 10th straight year. The Patriots trailed 24-10 at the end of the third quarter. They scored on a 40-yard field goal with 4:01 left in the game to pull within 26-10. Then, with 30 seconds remaining, Steve Lavelle scored on an 8-yard run and the Patriots added a two-point conversion to go ahead 27-26. The game ended on a Scottie Hunter interception. Quarterback Tom Brady threw an interception in the first half with a broken calf.

Classifier



Applicable NLP tasks:

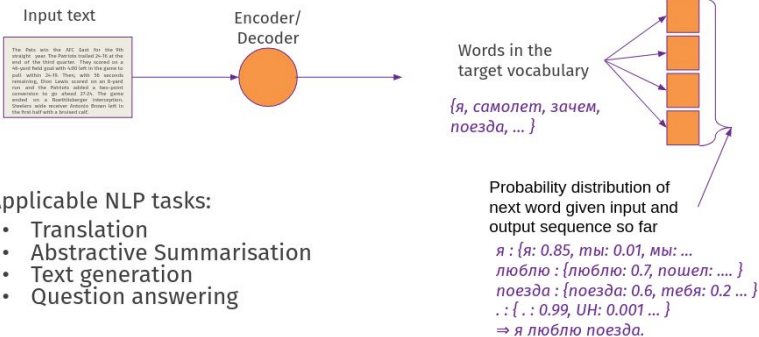
- Question Answering
- Relation Extraction

I : {start: 0.8, end: 0.01}  
like : {start: 0.1, end: 0.9}  
trains : {start: 0.05, end: 0.05}  
.: {start: 0.05, end: 0.04}  
⇒ [0, 1]: I

# Week 4 - Sequence modelling

## Sequence to sequence

*I like trains.*

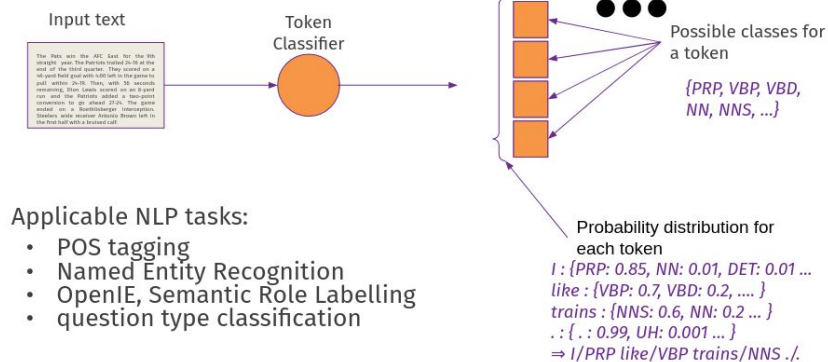


Applicable NLP tasks:

- Translation
- Abstractive Summarisation
- Text generation
- Question answering

## Sequence labelling

*I like trains.*



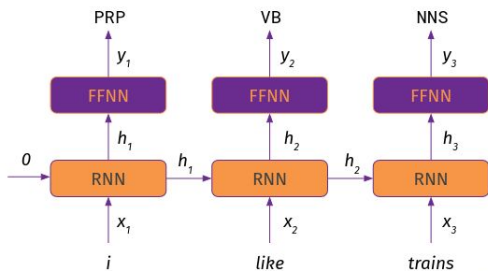
Applicable NLP tasks:

- POS tagging
- Named Entity Recognition
- OpenIE, Semantic Role Labelling
- question type classification



# Week 4: RNN, LSTM, BiLSTM

## RNN: Forward run



$$h_0 = 0$$

$$h_t = g(Uh_{t-1} + Wx_t + b)$$

$$y_t = f(Vh_t + b_z)$$

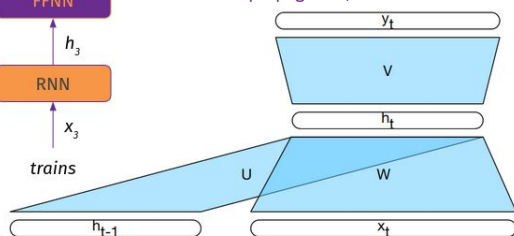
## RNN: vanishing gradients

$< 0.25!$  Lots of multiplications!

## RNN: Backward run

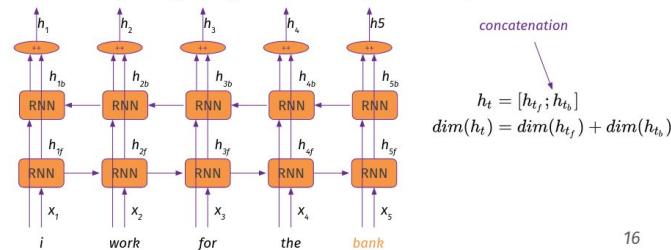
Backpropagation through time

$h_t$  depends on  $h_{t-1}$   
Unroll computation graph and do standard backpropagation, because  $t$  is not infinite.



## BiRNN

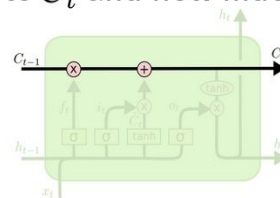
Idea: If we can go from left to right (“past”), why not also just go right to left (“future”)



16

## LSTM: Context vector

- Context, or memory vector  $C_t$  in addition to  $h_t$
- Context information from “the past” calculations
- At any step  $t$ , LSTM learns how much of  $h_t$  is to be “added” to  $C_t$  and how much of  $C_{t-1}$  is “kept”



For example:  
Information about grammatical gender of subject

# Week 4 - Contextualised embeddings

$d[\text{word}] = \text{vector}$   
 $f(\text{word}, \text{context}) = \text{contextualised\_vector}$

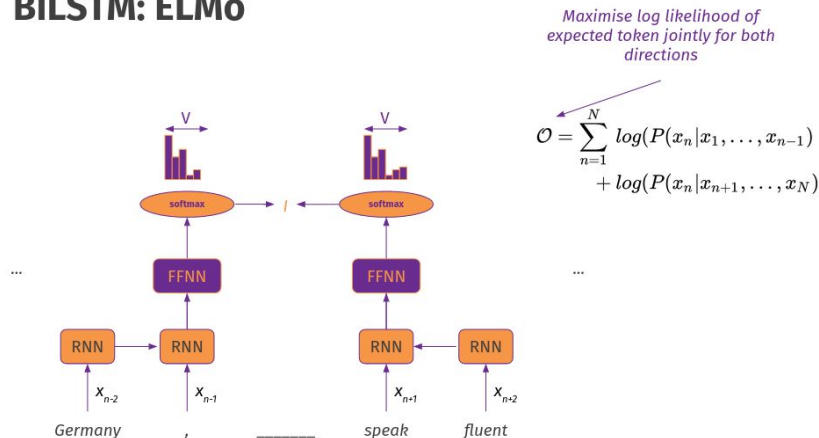
Task: probability of next word given  $n$  previous words.

$$P(x_{n+1} | x_1, \dots, x_n)$$

“I grew up in Germany, i speak fluent \_\_\_\_\_”

syntax information ✓  
semantic information ✓  
self supervision ✓

## Language modelling with BiLSTM: ELMo



# Week 4 - NER

## NER as a tagging problem (BIO scheme)

	Adam	Smith	works	for	IBM	in	London	.
POS tagging	NNP	NNP	VBZ	IN	NNP	IN	NNP	.
Entity recognition	B_PER	I_PER	O	O	B_ORG	O	B_LOC	O

**B**egin the mention

**I**nside the mention

- # classes = 2 \* # entity types + 1

# Week 4 - NER approaches

- Local approach: tags are *independent* each other
  - Any classifiers can be used
    - SVM: truly local
    - RNN, LSTM, BiLSTM: not truly local
- Global approach: tags are *dependent* each other
  - Hidden Markov Model (HMM)
  - Conditional Random Fields (CRF)

# Week 4: CRF vs. Neural networks

## CRF

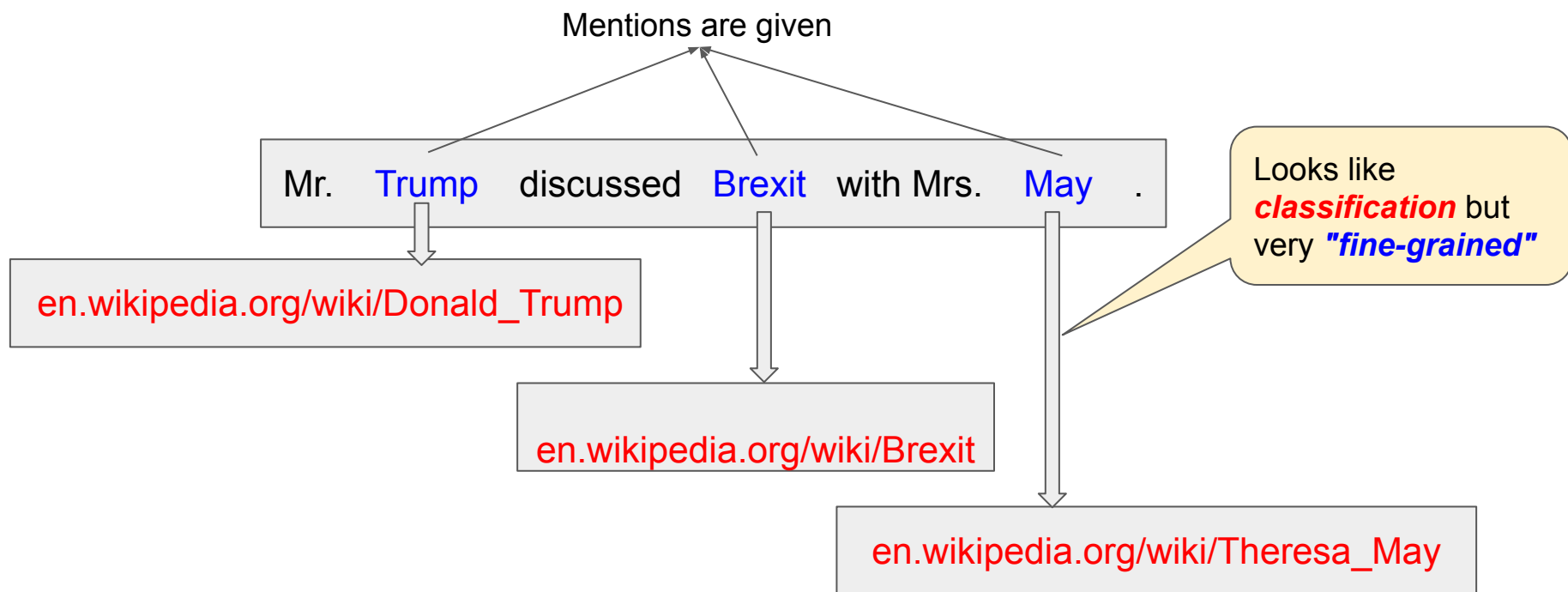
- Feature engineering
- Do not need pre-trained vectors
- Models are roughly interpretable
- *Perform well with datasets that have many NE categories(\*)*

## Neural networks

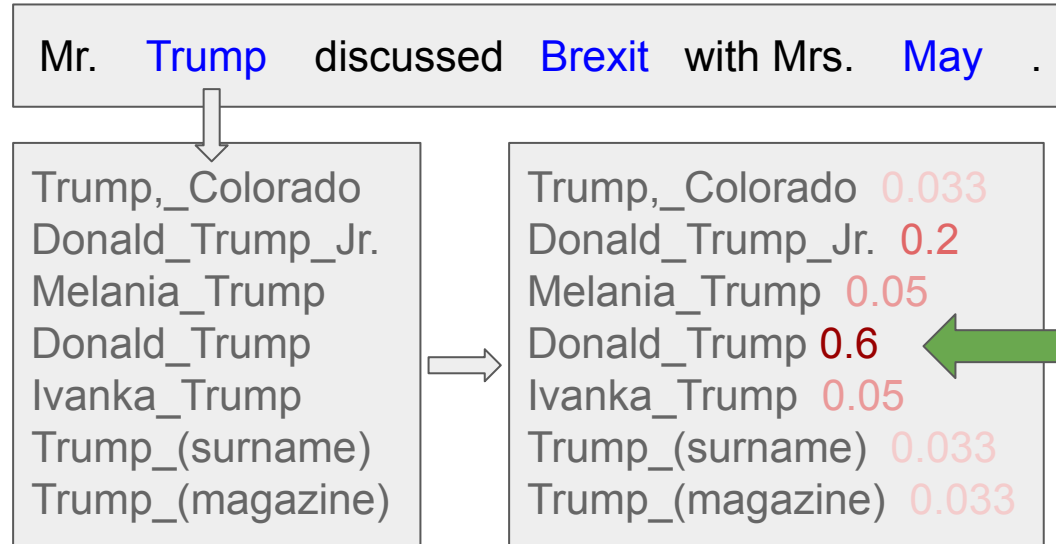
- Do not need features
- Need pre-trained vectors from big language models
- Models use implicit features (created by hidden layers) → not easy to interpret
- *Perform not so well with datasets that have many NE categories(\*)*

(\*) This observation is only based on my personal experiences

# Week 4 - NEL



# Week 4 - NEL basic steps



***Candidate generation***  
(A short list of  
candidates, < 100)

***Candidate ranking***

Thank you very much  
Good luck with your exams!