# Natural language processing with neural networks.

Hubert Bryłkowski Europython 2019

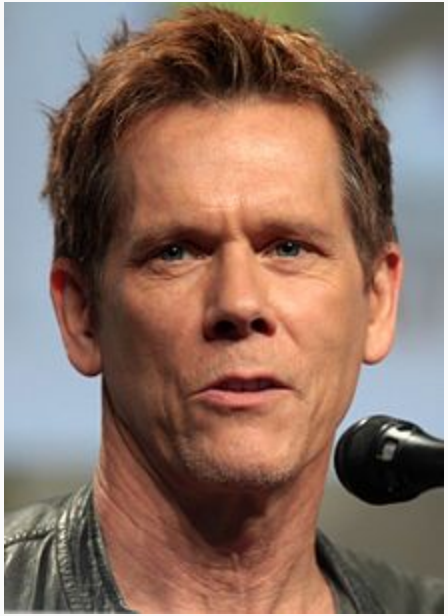
# Hubert Bryłkowski

hubert@brylkowski.com
linkedin.com/in/hubert-bry%C5%82kowski/

# Why NLP is hard

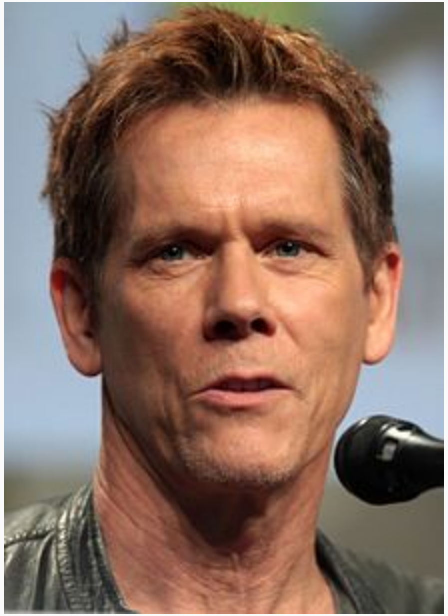
# Ambiguity

I had a sandwich with Bacon.



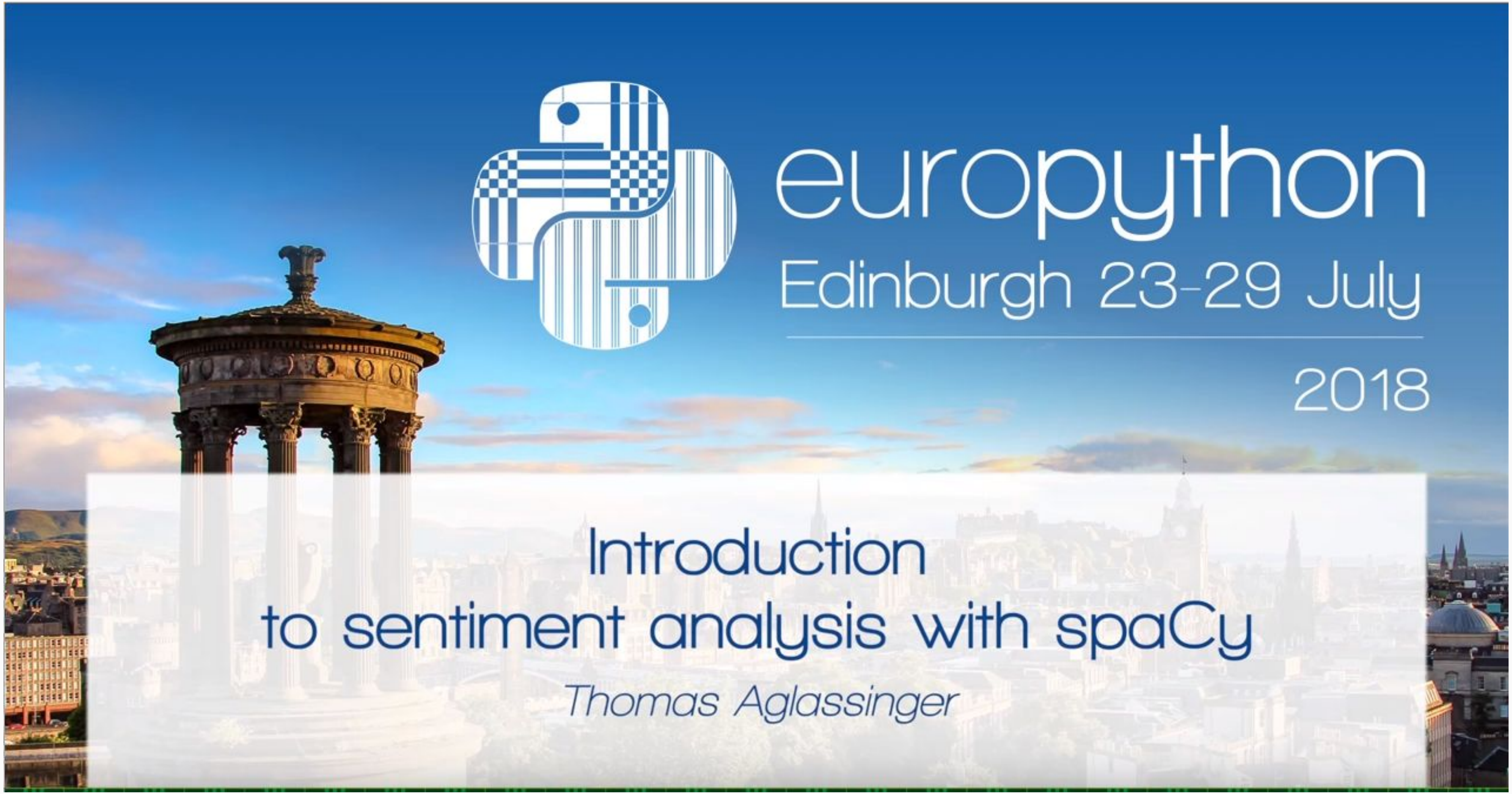
# Ambiguity

I had a sandwich with Bacon.



# Texts are compositional

Characters -> words -> sentences -> paragraphs



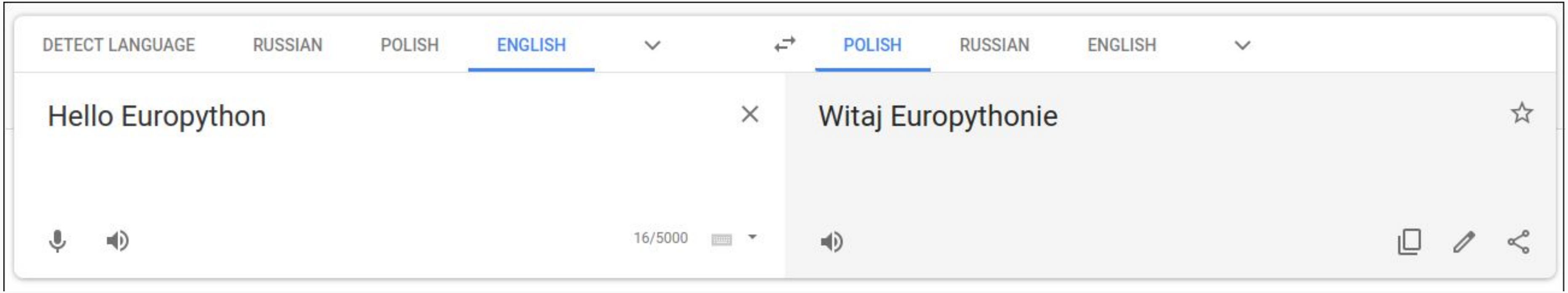

https://www.youtube.com/watch?v=LvlUBxi_JEg

# Common problems in NLP

Document classification (sentiment, author, spam)

# Common problems in NLP

Sequence to sequence (translation, summarization, response generation)



# Common problems in NLP

Information extraction (named-entity recognition)

Jimmy bought Apple shares.

company

Jimmy bought an apple.

fruit

# Why neural networks are good for NLP?

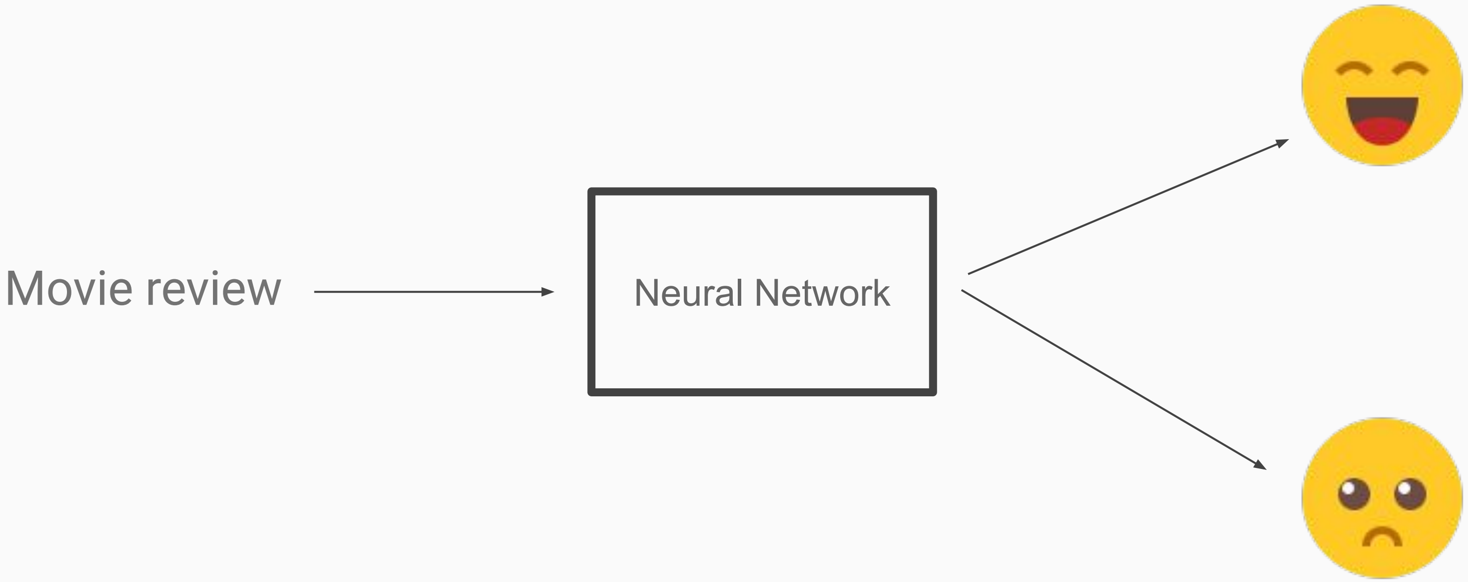
# “Real” life problem

# IMDB sentiment analysis.

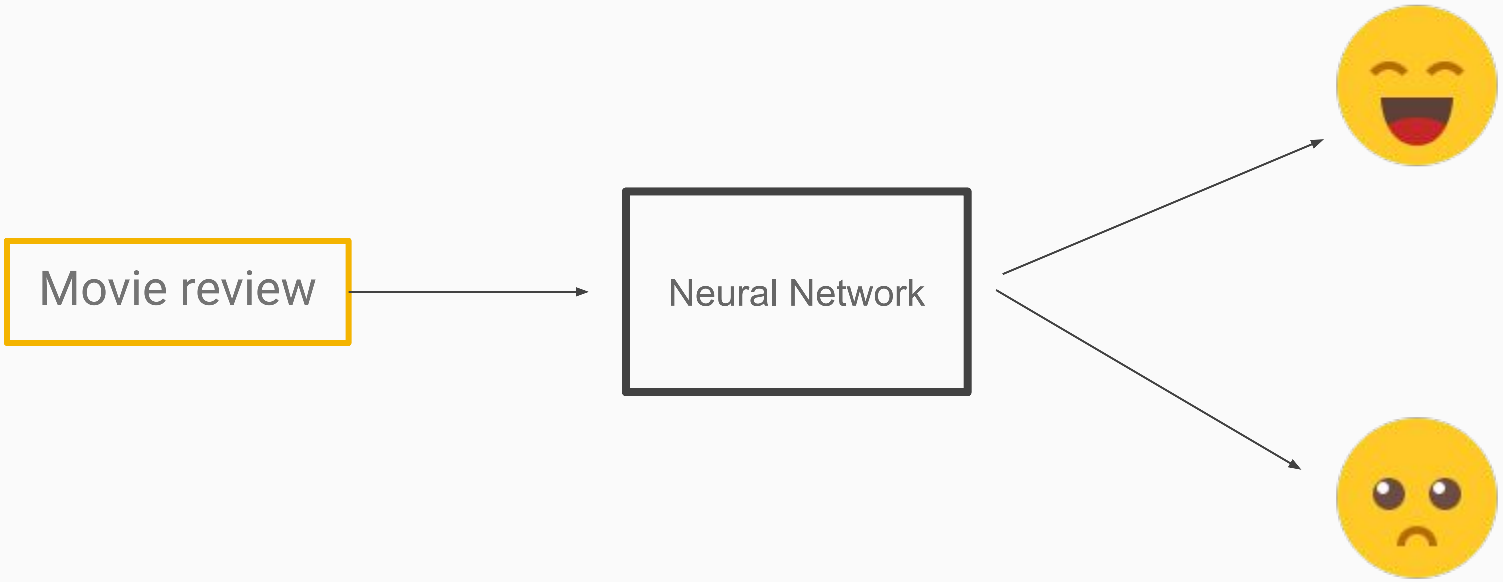
## 25,000 highly polar movie reviews

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis. *The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).*

# Task definition



# Task definition



# Text as input

“A big disappointment for what was touted as an incredible film. Incredibly bad. Very pretentious. It would be nice if just once someone would create a high profile role for a young woman that was not (...)”

# Possible features

A quick brown fox.

# Possible features

A quick brown **fox**.

# Possible features

A quick brown **fox**.

noun

# Possible features

A quick brown **fox**.

noun
canine

# Possible features

A quick brown **fox**.

noun
canine
stem - fox
lemma - fox

# Possible features

A quick brown **fox**.

noun

canine

stem - fox

lemma - fox

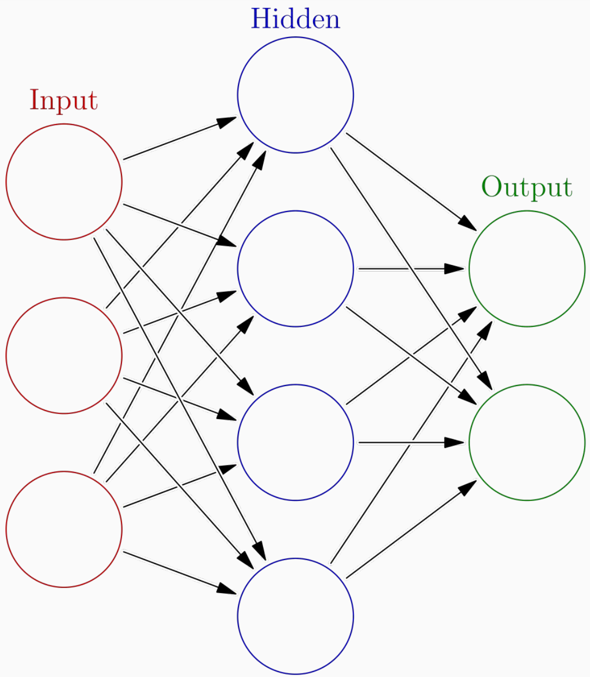
TFIDF

# Bag of words

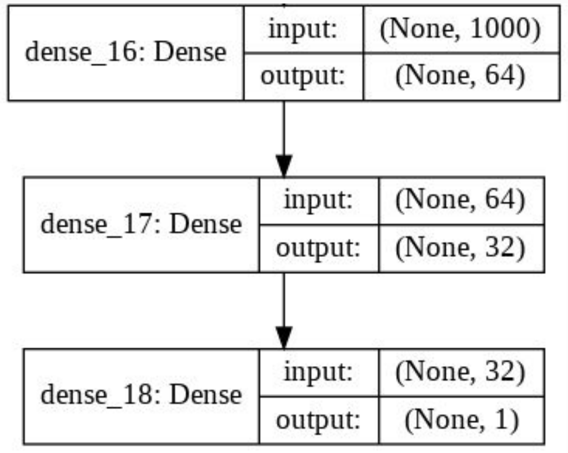
A quick brown fox.

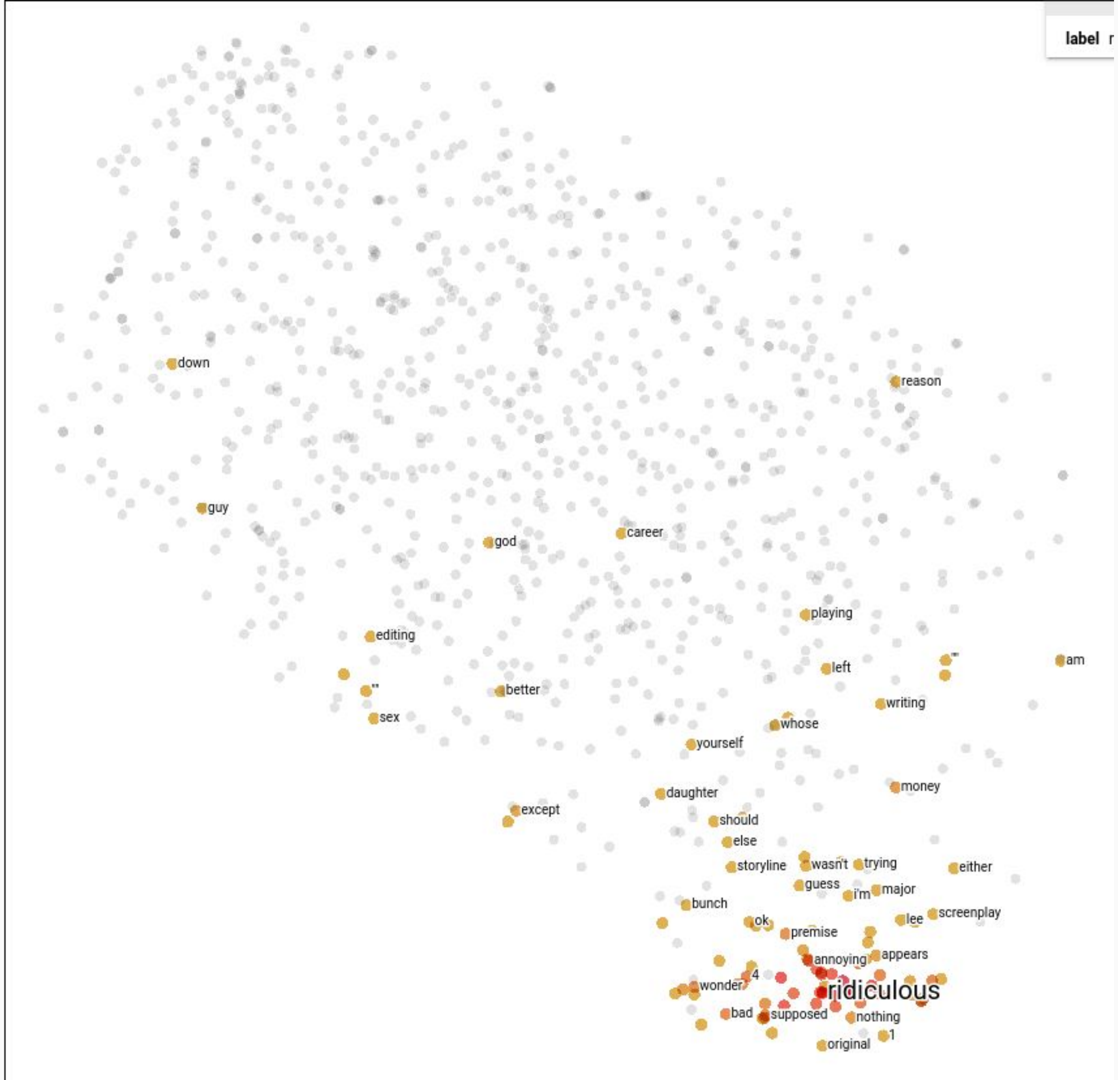
| vocab | X |
|---|---|
| fox | 1 |
| brown | 1 |
| over | 0 |
| quick | 1 |
| a | 1 |
| jumps | 0 |
| dog | 0 |
| lazy | 0 |
| <UNK> | 0 |

# Fully connected neural network



# Simple model






Nearest points in the original space:

| | |
|---|---|
| waste | 0.255 |
| boring | 0.298 |
| worst | 0.301 |
| worse | 0.309 |
| poorly | 0.311 |
| horrible | 0.336 |
| annoying | 0.339 |
| awful | 0.347 |
| poor | 0.356 |
| terrible | 0.361 |
| crap | 0.377 |


Nearest points in the original space:

| | |
|---|---|
| excellent | 0.306 |
| 7 | 0.371 |
| simple | 0.403 |
| 8 | 0.407 |
| amazing | 0.409 |
| favorite | 0.413 |
| today | 0.442 |
| highly | 0.444 |
| wonderful | 0.458 |
| superb | 0.458 |

# Pros and cons of FC with BoW

- Simple - cheap and fast to train
- Always looking at whole text
- Kinda interpretable

- Can’t get close to state of the art
- Order of words do not matter

# Bag of words

I loved the movie, but cinema was terrible.

I loved cinema, but the movie was terrible.

# Sequence of one-hot vectors

A quick brown fox.

| **vocab** | **X** | | | |
|---|---|---|---|---|
| fox | 0 | 0 | 0 | **1** |
| brown | 0 | 0 | **1** | 0 |
| over | 0 | 0 | 0 | 0 |
| quick | 0 | **1** | 0 | 0 |
| a | **1** | 0 | 0 | 0 |
| jumps | 0 | 0 | 0 | 0 |
| dog | 0 | 0 | 0 | 0 |
| lazy | 0 | 0 | 0 | 0 |
| <UNK> | 0 | 0 | 0 | 0 |

# Sequence of one-hot vectors

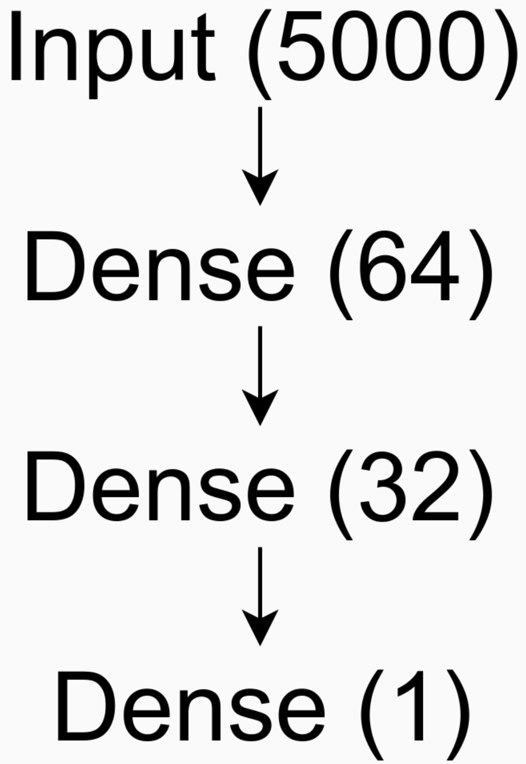
A quick brown vixen.

| vocab | X | | | |
|---|---|---|---|---|
| fox | 0 | 0 | 0 | 0 |
| brown | 0 | 0 | **1** | 0 |
| over | 0 | 0 | 0 | 0 |
| quick | 0 | **1** | 0 | 0 |
| a | **1** | 0 | 0 | 0 |
| jumps | 0 | 0 | 0 | 0 |
| dog | 0 | 0 | 0 | 0 |
| lazy | 0 | 0 | 0 | 0 |
| <UNK> | 0 | 0 | 0 | **1** |

# Sequence of one-hot vectors

A quick brown vixen.

| **vocab** | **X** | | | |
|---|---|---|---|---|
| fox | 0 | 0 | 0 | 0 |
| brown | 0 | 0 | **1** | 0 |
| over | 0 | 0 | 0 | 0 |
| quick | 0 | **1** | 0 | 0 |
| a | **1** | 0 | 0 | 0 |
| jumps | 0 | 0 | 0 | 0 |
| dog | 0 | 0 | 0 | 0 |
| lazy | 0 | 0 | 0 | 0 |
| <NOUN> | 0 | 0 | 0 | **1** |
| <ADJ> | 0 | 0 | 0 | 0 |


Input (5000)
Dense (64)
Dense (32)
Dense (1)

# Sequence of one-hot vectors

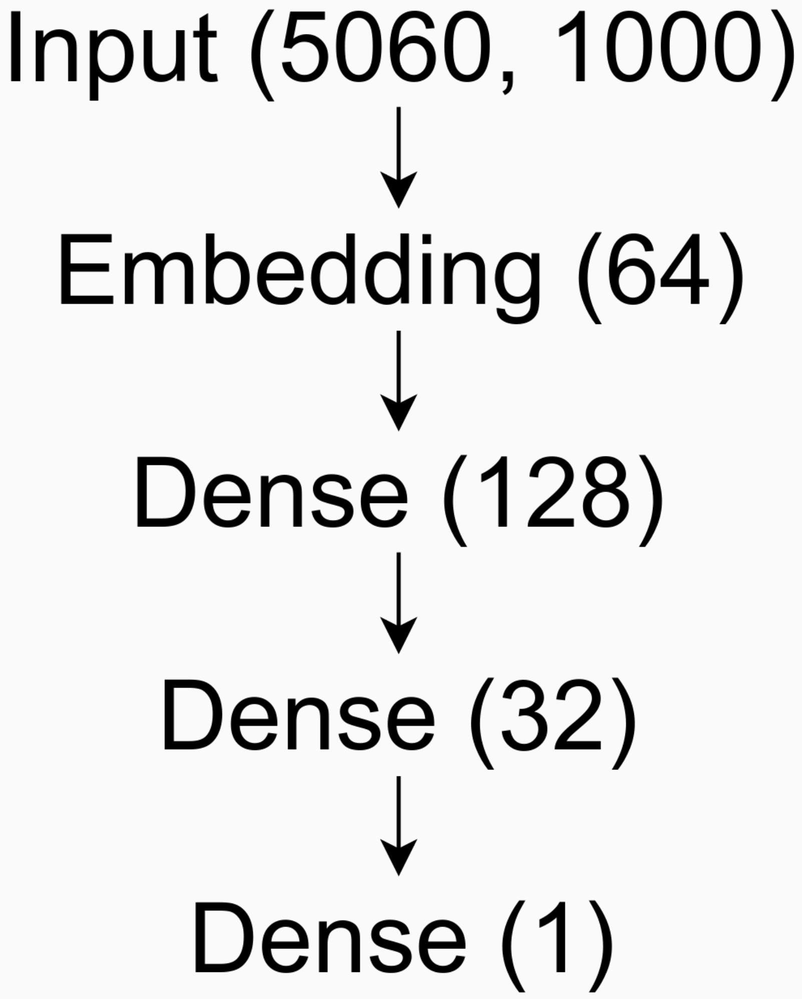
A quick brown vixen.

| **vocab** | **X** | | | |
|---|---|---|---|---|
| fox | 0 | 0 | 0 | 0 |
| brown | 0 | 0 | **1** | 0 |
| over | 0 | 0 | 0 | 0 |
| quick | 0 | **1** | 0 | 0 |
| a | **1** | 0 | 0 | 0 |
| lazy | 0 | 0 | 0 | 0 |
| <UNK> | 0 | 0 | 0 | **1** |
| <NOUN> | 0 | 0 | 0 | **1** |
| <ADJ> | 0 | **1** | **1** | 0 |
| <DET> | **1** | 0 | 0 | 0 |

# Sequence of embeddings

A quick brown vixen.

| vocab | X | | | |
|---|---|---|---|---|
| word | 0.01 | 0.84 | -0.54 | 0.03 |
| | 0.18 | 0.96 | -0.45 | 0.98 |
| | -0.63 | -0.21 | -0.82 | -0.60 |
| | 0.94 | -0.37 | 0.72 | 0.69 |
| Part of speech | 0.20 | -0.38 | 0.90 | 0.11 |
| | 0.43 | 0.70 | -0.91 | -0.97 |


Input (5060, 1000)
Embedding (64)
Dense (128)
Dense (32)
Dense (1)

# Pros and cons of FC with sequence

- Still simple - cheap and fast to train
- Order of words matter
- Kinda interpretable

- Can’t get close to state of the art (0.96 - GraphStar)
- Words at given position matter more
- Negations are hard to catch

[Deep learning course - Andrew Ng]

This movie was not good.

This movie was not_good.

# Convolutional Neural Networks - CNNs

This movie was not good

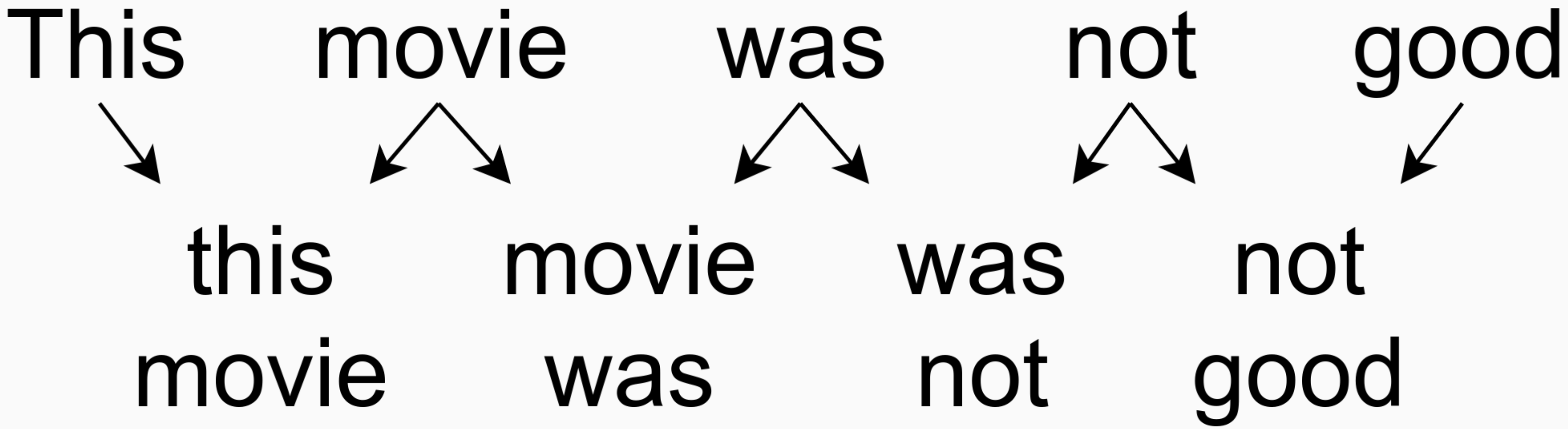
This movie was not good

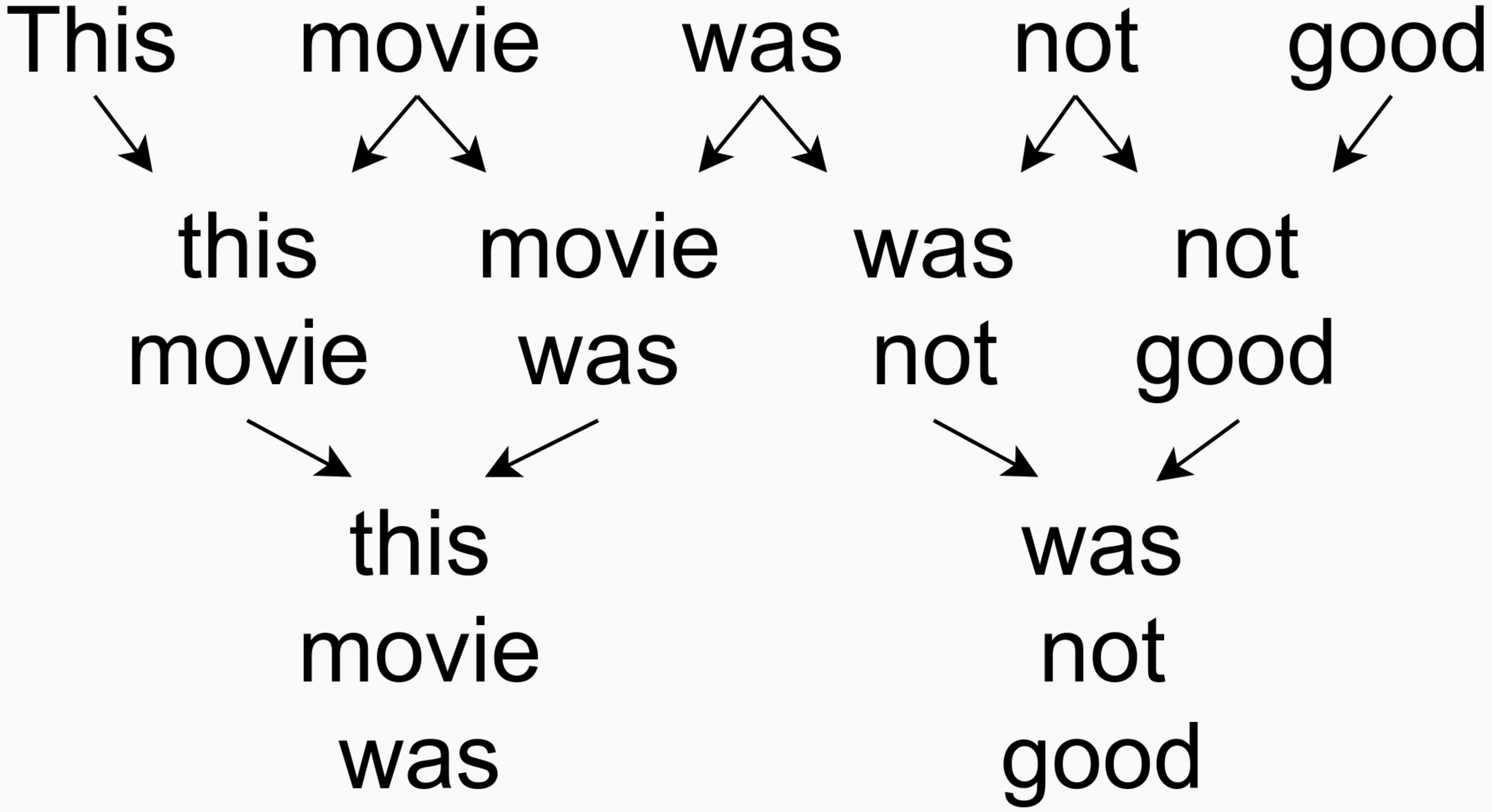
this
movie

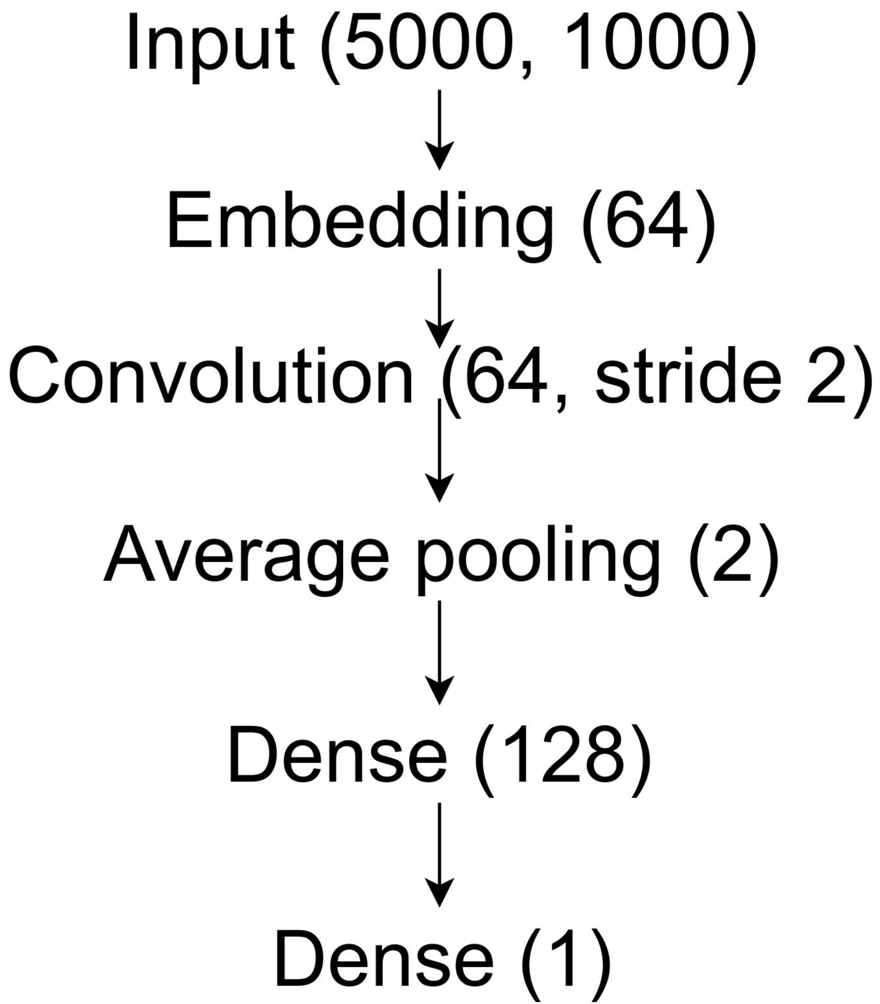
This movie was not good

this movie

movie was


This movie was not good
this movie
movie was
was not
not good


This
movie
was
not
good
this movie
movie was
was not
not good
this movie was
was not good


Input (5000, 1000)
Embedding (64)
Convolution (64, stride 2)
Average pooling (2)
Dense (128)
Dense (1)

# Pros and cons of CNNs

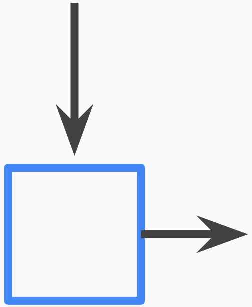
- Parallelize nicely - inference can be fast
- Order of words matter
- Positions of words matter
- We can look at whole sentence

- Connections can only be made between close neighbours

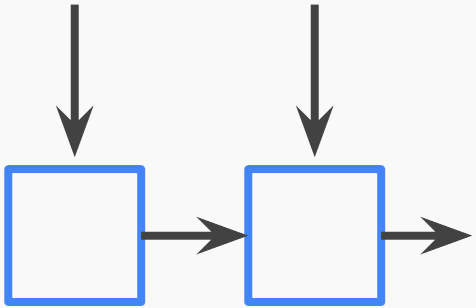
[Understanding Convolutional Neural Networks for NLP - DENNY BRITZ]

# Recurrent Neural Networks - RNNs

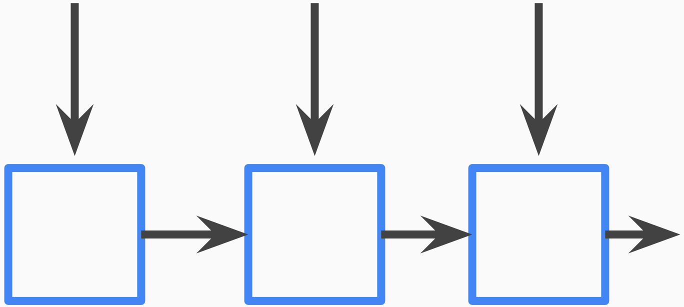
This movie was not good.

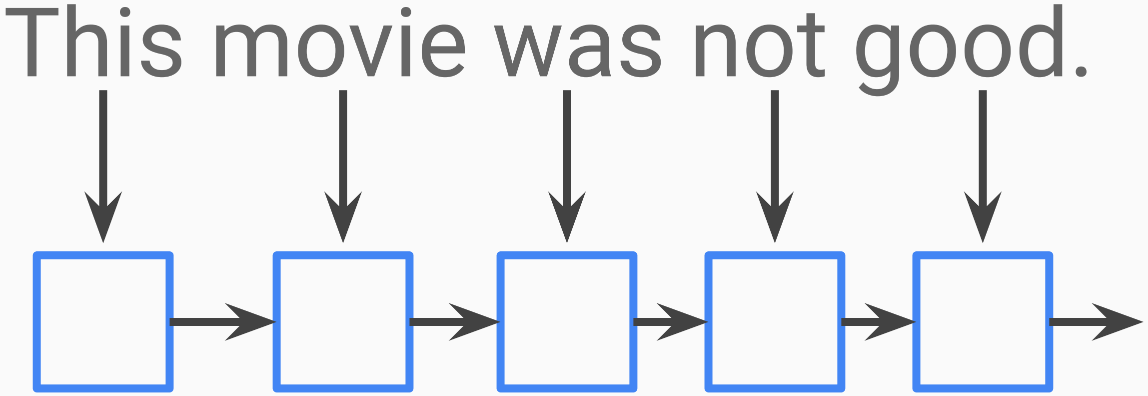


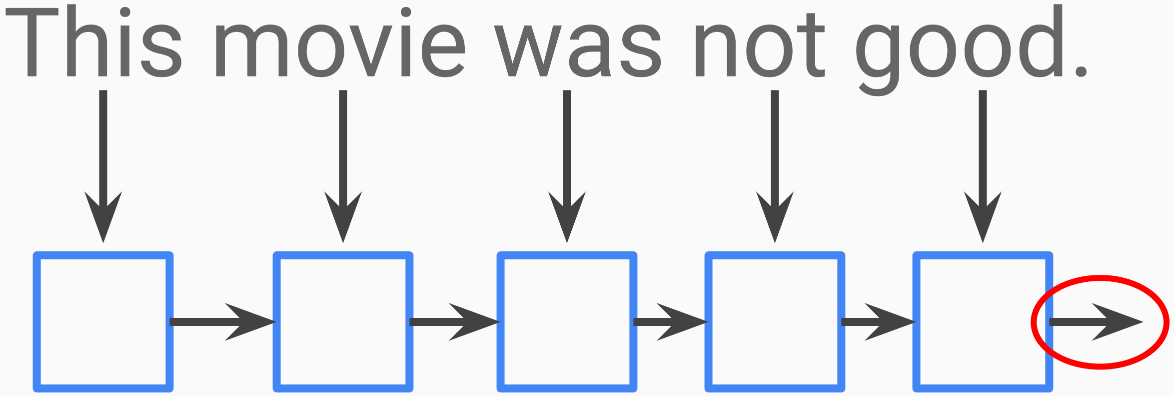
This movie was not good.

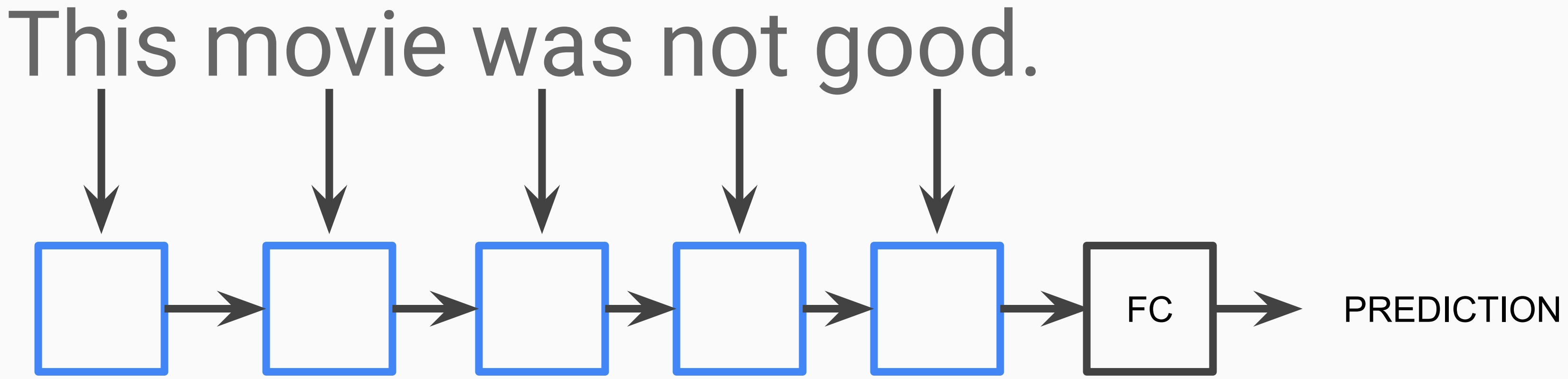


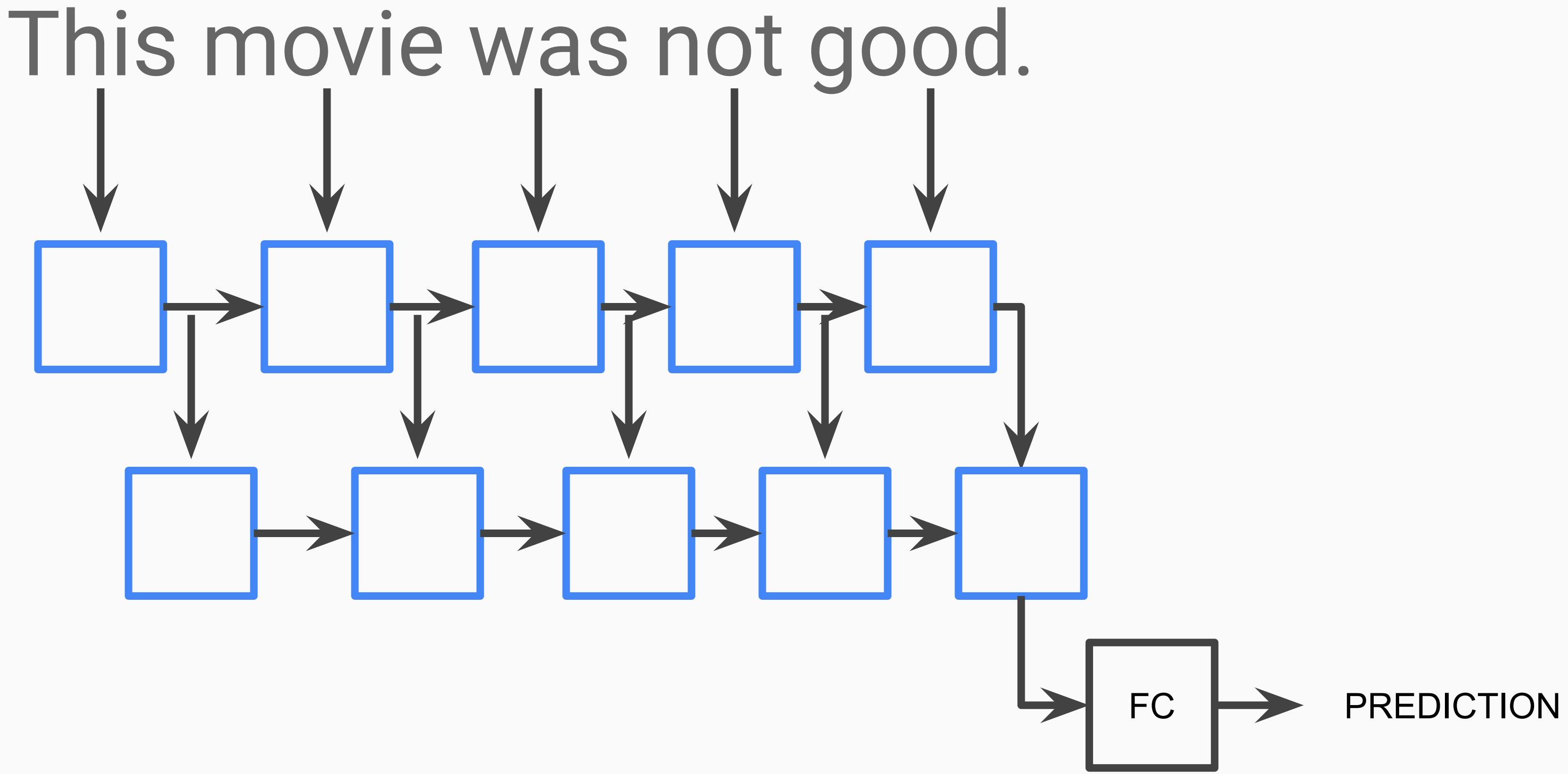
This movie was not good.

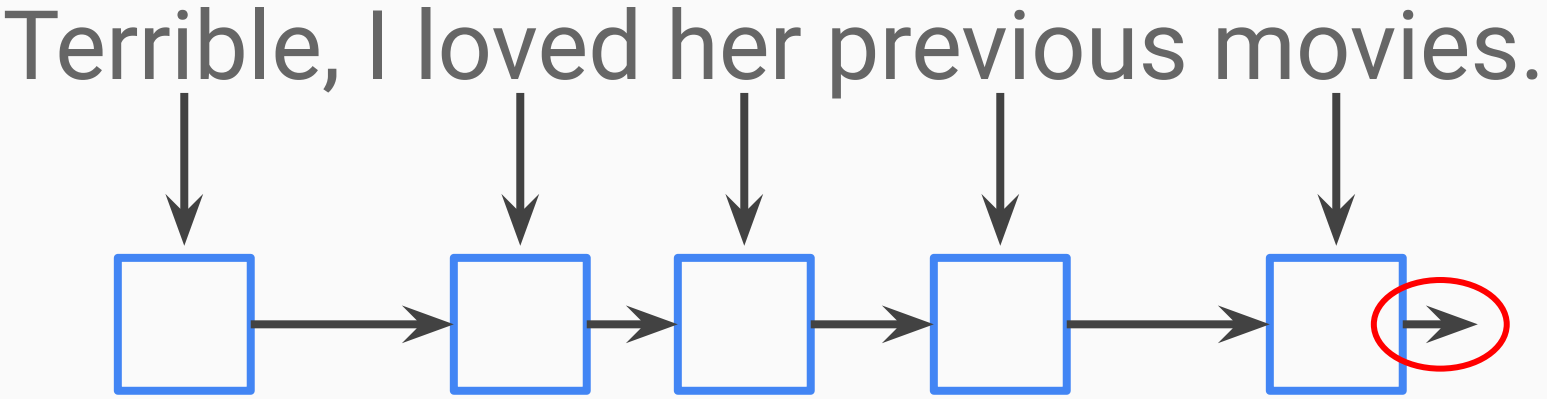


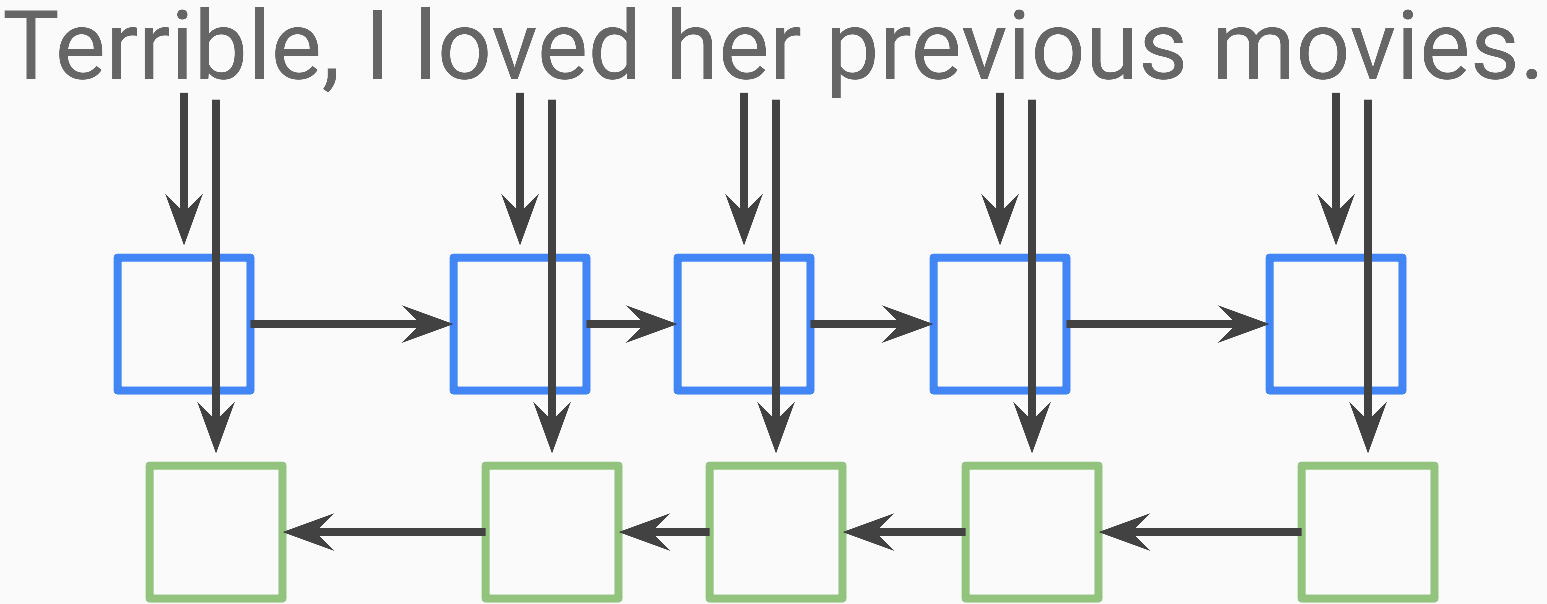

This movie was not good.

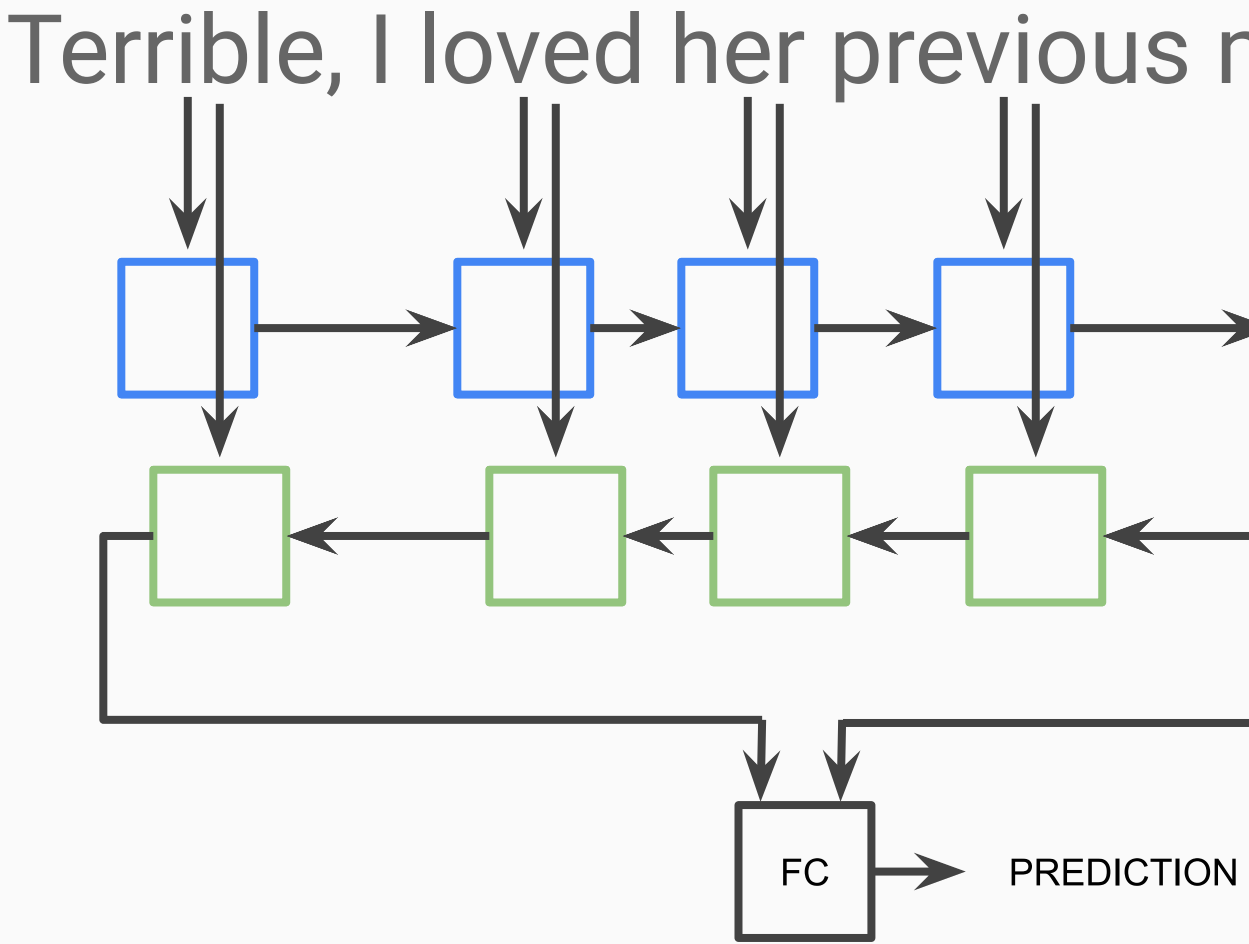

This movie was not good.


This movie was not good.
FC
PREDICTION


This movie was not good.
FC
PREDICTION


Terrible, I loved her previous movies.


Terrible, I loved her previous movies.


Terrible, I loved her previous movies.
FC
PREDICTION

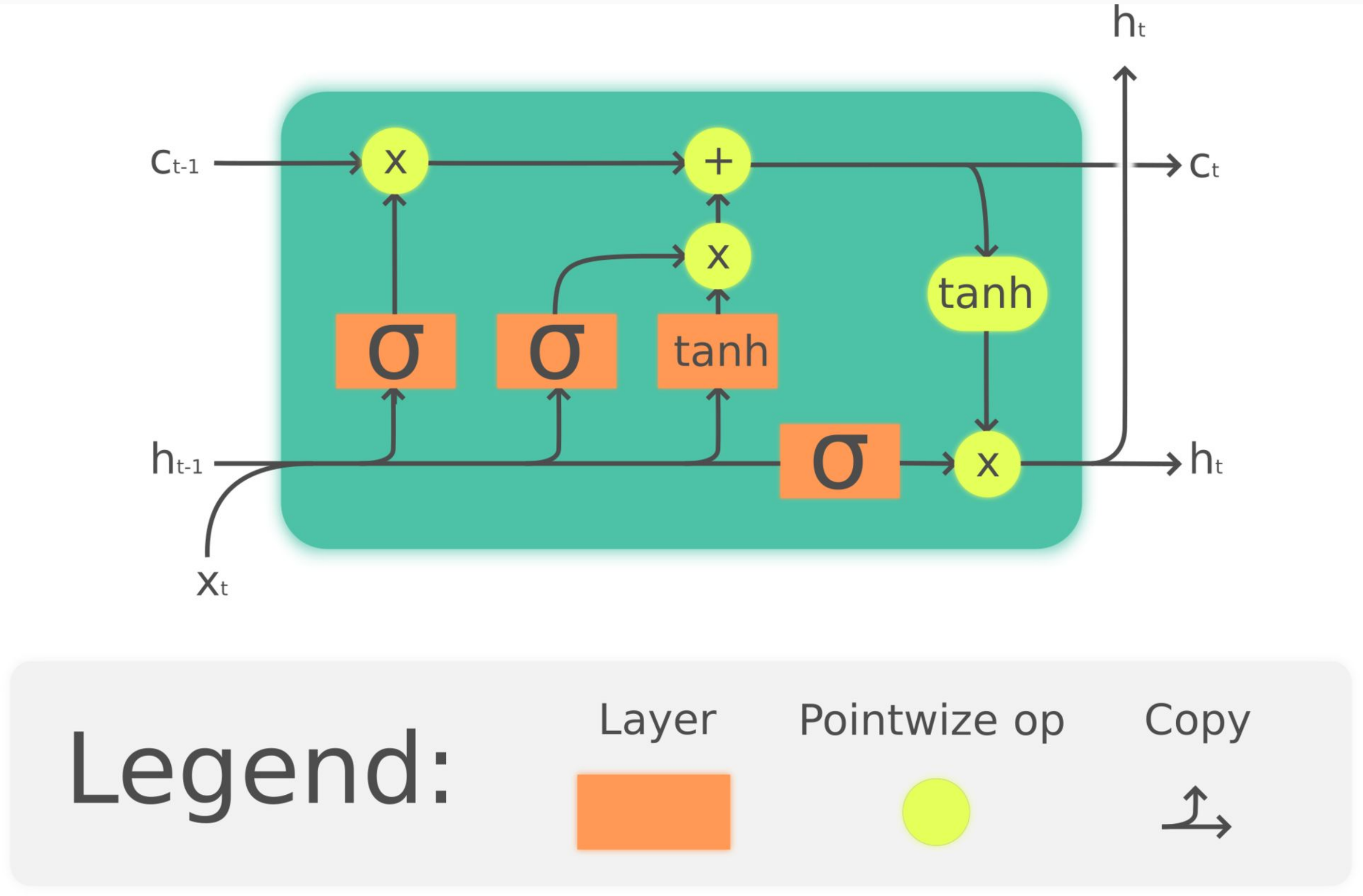
# Pros and cons of simple RNNs

- Can give better results
- We look at whole sequence

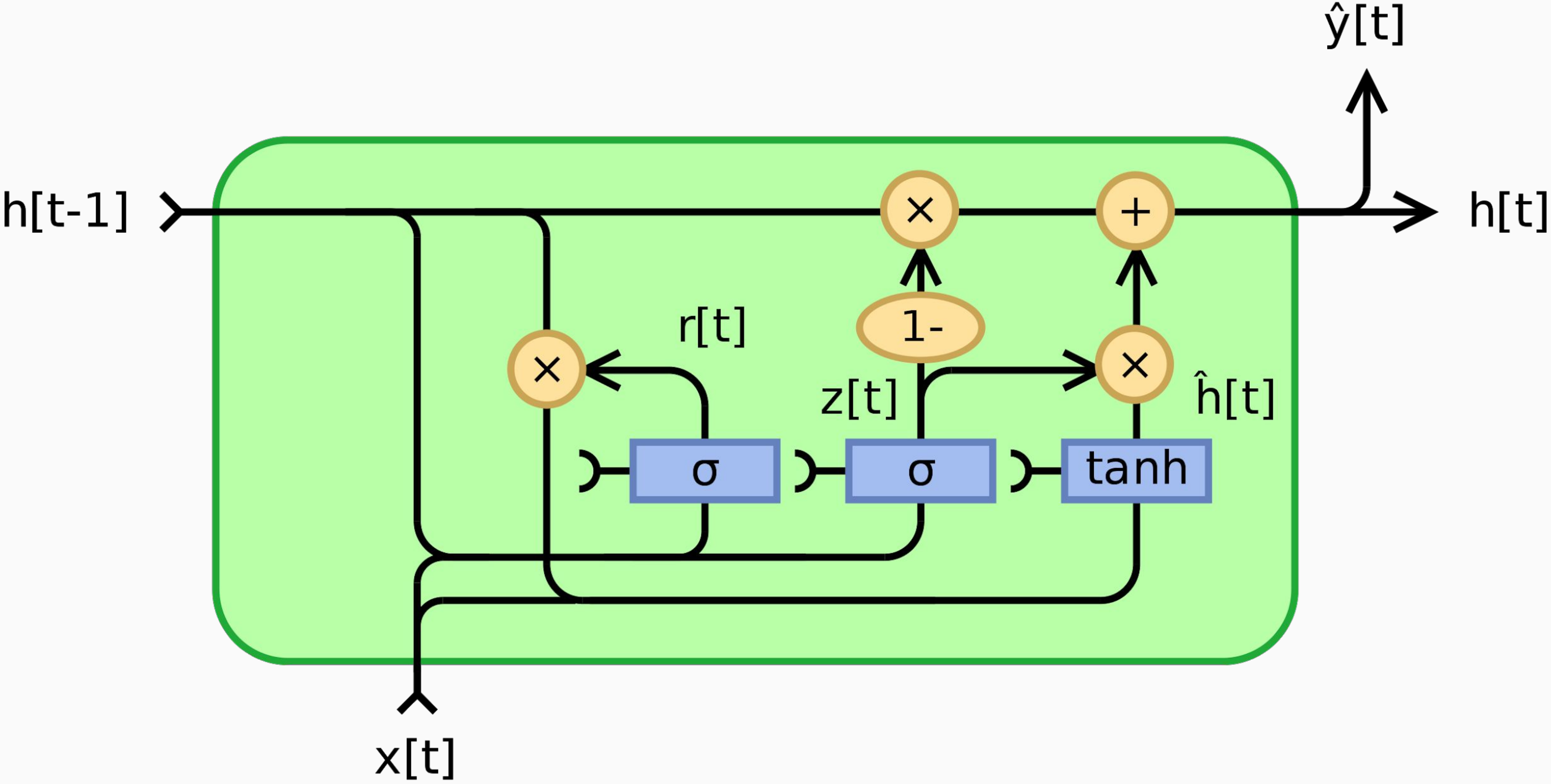
- Hard to train - a lot of resources and time needed
- Prone to “forgetting” words from beginning (or end) of sequence

[Stanford lecture Recurrent Neural Networks and Language Models]

# LSTM / GRU



$h_t$
$c_{t-1}$
x
+
$c_t$
x
tanh
σ
σ
tanh
$h_{t-1}$
σ
x
$h_t$
$x_t$
Legend:
Layer
Pointwize op
Copy


ŷ[t]
h[t-1]
×
+
h[t]
r[t]
1-
×
×
z[t]
ĥ[t]
σ
σ
tanh
x[t]

# Pros and cons of LSTM / GRU

- Can give best results
- Always look at whole sequence
- Can “remember” the words from beginning

- Hardest to train - a lot of resources and time needed
- Not counting transformer - best models

[Stanford lecture Machine Translation and Advanced Recurrent LSTMs and GRUs]
[Understanding LSTM Networks]

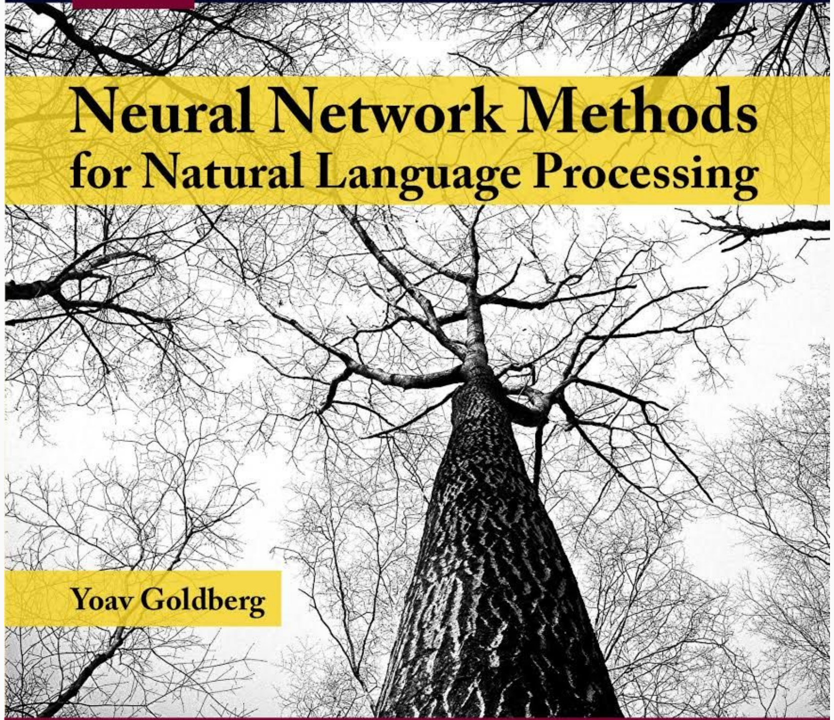
# Summary

| architecture | accuracy | 1 epoch time |
|---|---|---|
| fully Connected with bow | 0.89 | 2s |
| fully connected - embeddings | 0.89 | 1s |
| fully connected - pos instead unk | 0.88 | 5s |
| fully connected - pos embeddings | 0.88 | 3s |
| simple RNN - embeddings | 0.85 | 42s |
| simple biRNN - embeddings | 0.87 | 137s |
| LSTM | 0.88 | 137s |

https://colab.research.google.com/drive/1J3VyPNiLQ-SpA_HBw29HRjv8Oa1Ls3zJ

MORGAN & CLAYPOOL PUBLISHERS

# Neural Network Methods for Natural Language Processing



Yoav Goldberg

*SYNTHESIS LECTURES ON HUMAN LANGUAGE TECHNOLOGIES*

Graeme Hirst, *Series Editor*

# Thank you

hubert@brylkowski.com

linkedin.com/in/hubert-bry%C5%82kowski/