PSY9511: Seminar 8

Sequence modelling (with an emphasis on language)

Esten H. Leonardsen

13.05.24

UNIVERSITETET
I OSLO

Overview

1. Introduction and motivation
2. Preprocessing
3. Bag of words
4. Vectorization
5. Recurrent neural networks
6. Transformers

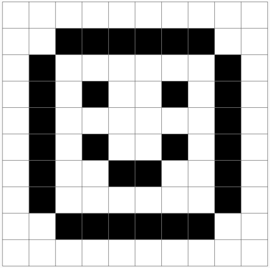
Introduction

Introduction

Age	Sex	Education	Salary
25	Male	12	40,000
30	Female	16	65,000
35	Male	14	55,000
40	Female	18	80,000
45	Male	16	75,000

Introduction

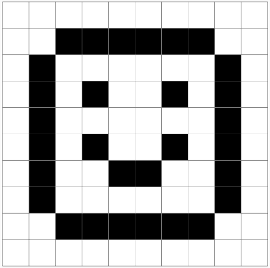
Age	Sex	Education	Salary
25	Male	12	40,000
30	Female	16	65,000
35	Male	14	55,000
40	Female	18	80,000
45	Male	16	75,000



Introduction

The movie was great, the actors were awesome.

Age	Sex	Education	Salary
25	Male	12	40,000
30	Female	16	65,000
35	Male	14	55,000
40	Female	18	80,000
45	Male	16	75,000



Introduction

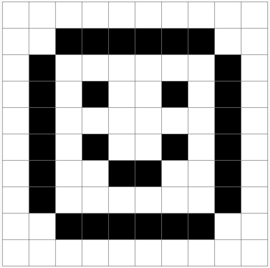
The movie was great, the actors were awesome.

Introduction

The movie was great, the actors were awesome.

The movie was great, the actors were awesome.

Age	Sex	Education	Salary
25	Male	12	40,000
30	Female	16	65,000
35	Male	14	55,000
40	Female	18	80,000
45	Male	16	75,000



Introduction

Age	Sex	Education	Salary
25	Male	12	40,000
30	Female	16	65,000
35	Male	14	55,000
40	Female	18	80,000
45	Male	16	75,000

Introduction

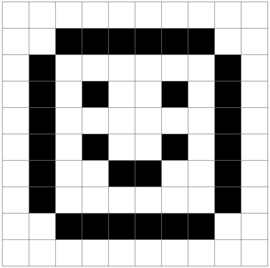
Age

35

Introduction

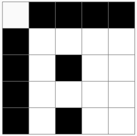
Age	Sex
35	Male

Introduction

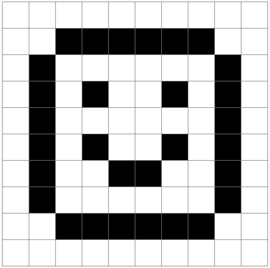


Introduction

Introduction



Introduction



The movie was great, the actors were awesome.

Introduction

movie

Introduction

The movie was great, the actors were awesome.

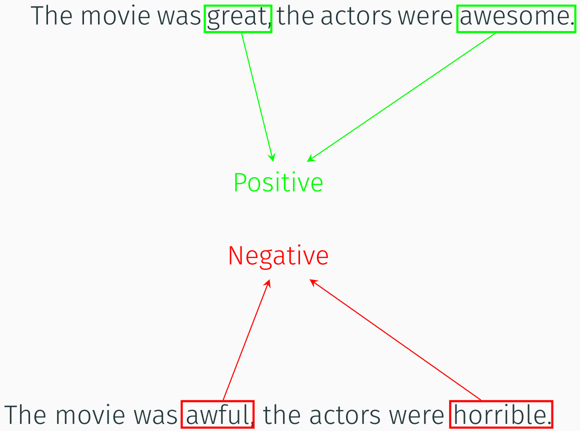
Introduction

The movie was great, the actors were awesome.

Positive

Negative

Introduction



Introduction

The movie was great, the actors were awesome.

La película fue genial, las actores fueron increíbles.

Introduction

The movie was great, the actors were awesome.

La película fue genial, las actores fueron increíbles.

Introduction

The movie was great, the actors were _________ .

Introduction

Introduction

The movie was _____, the actors were _________ .

Introduction

The movie was great, the actors were __________ .

Introduction

The movie was great, the actors were awesome.

Introduction

The movie was $\boxed{\text{great,}}$ the actors were $\boxed{\text{awesome.}}$

Introduction

The movie was $\boxed{\text{great,}}$ we saw it at the new Cinema in the city center, the actors were $\boxed{\text{awesome.}}$

Introduction

The movie was $\boxed{\text{great,}}$ we saw it at the new Cinema in the city center, right down by the restaurant where we went for my birthday that one year, the one where the clown was inside the cake, the actors were $\boxed{\text{awesome.}}$

Introduction

Language modelling: Using the innate structure in language to create better models

- Classification: Predict a class for a full sequence (sentiment analysis)
- Sequence-to-sequence: Predict a sequence from another sequence (translation)
- Generative: Predict the next token in a sequence of words

Preprocessing

The movie was great, the actors were awesome.

Preprocessing

The movie was great, the actors were awesome.

["the" "movie" "was" "great" "," "the" "actors" "were" "awesome" "."]

Tokenization

Preprocessing

The movie was great, the actors were awesome.

[<s> ”the” ”movie” ”was” ”great” ”,” ”the” ”actors” ”were” ”awesome” ”.” <e>]

Tokenization

Preprocessing

The movie was great, the actors were awesome.

[<s> "the" "movie" "was" "great" "," "the" "actors" "were" "awesome" "." <e>]

In[1]:
```python
from nltk.tokenize import word_tokenize

tokens = word_tokenize(s)
tokens = [token.lower() for token in tokens]
tokens = ['<s>'] + tokens + ['<e>']
print(tokens)
```

Out[1]:
```
['<s>', 'the', 'movie', 'was', 'great', ',', 'the', 'actors',
'were', 'awesome', '.', '<e>']
```

Tokenization

Preprocessing

The movie was great, the actors were awesome.

[<s> "the" "movie" "was" "great" "," "the" "actors" "were" "awesome" "." <e>]

[<s> "The" "movie" "was" "great" "," "the" "actor" "were" "awesome" "." <e>]

Stemming

The movie was great, the actors were awesome.

[<s> "the" "movie" "was" "great" "," "the" "actors" "were" "awesome" "." <e>]

[<s> "The" "movie" "was" "great" "," "the" "actor" "were" "awesome" "." <e>]

In[1]:
```python
from nltk.stem.snowball import SnowballStemmer

stemmer = SnowballStemmer('english')
stemmed = [stemmer.stem(token) for token in tokens]
stemmed
```

Out[1]:
```
['<s>', 'the', 'movi', 'was', 'great', ',', 'the', 'actor',
'were', 'awesom', '.', '<e>']
```

Stemming

Preprocessing

The movie was great, the actors were awesome.

[<s> ”the” ”movie” ”was” ”great” ”,” ”the” ”actors” ”were” ”awesome” ”.” <e>]

[<s> ”The” ”movie” ”was” ”great” ”,” ”the” ”actor” ”was” ”great” ”.” <e>]

Lemmatization

Preprocessing

The movie was great, the actors were awesome.

[<s> "the" "movie" "was" "great" "," "the" "actors" "were" "awesome" "." <e>]

[<s> "The" "movie" "was" "great" "," "the" "actor" "was" "great" "." <e>]

In[1]:
```python
from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()
lemmatized = [lemmatizer.lemmatize(token) for token in tokens]
print(lemmatized)
```

Out[1]:
```
['<s>', 'the', 'movie', 'wa', 'great', ',', 'the', 'actor',
'were', 'awesome', '.', '<e>']
```

Lemmatization

Preprocessing

The movie was great, the actors were awesome.

[<s> ”the” ”movie” ”was” ”great” ”,” ”the” ”actors” ”were” ”awesome” ”.” <e>]

[<s> ”movie” ”great” ”,” ”actor” ”awesome” ”.” <e>]

Stopword removal

Preprocessing

The movie was great, the actors were awesome.

[<s> "the" "movie" "was" "great" "," "the" "actors" "were" "awesome" "." <e>]

[<s> "movie" "great" "," "actor" "awesome" "." <e>]

In[1]:

```python
from nltk.corpus import stopwords

pruned = [token for token in tokens if not token in stopwords.
      words('english')]
print(pruned)
```

Out[1]:

```
['<s>', 'movie', 'great', ',', 'actors', 'awesome', '.', '<e>']
```

Stopword removal

Preprocessing

The movie was great, the actors were awesome.

[<s> "the" "movie" "was" "great" "," "the" "actors" "were" "awesome" "." <e>]

["," "." <e> <s> "actors" "awesome" "great" "movie" "the" "was" "were"]
0 1 2 3 4 5 6 7 8 9 10

Preprocessing

The movie was great, the actors were awesome.

[<s> ”the” ”movie” ”was” ”great” ”,” ”the” ”actors” ”were” ”awesome” ”.” <e>]

[3 8 7 9 6 0 8 4 10 5 1 2]

[”,” ”.” <e> <s> ”actors” ”awesome” ”great” ”movie” ”the” ”was” ”were”]

0 1 2 3 4 5 6 7 8 9 10

Integer encoding

Preprocessing

The movie was great, the actors were awesome.

[<s> "the" "movie" "was" "great" "," "the" "actors" "were" "awesome" "." <e>]

[3 8 7 9 6 0 8 4 10 5 1 2]

Integer encoding

Preprocessing

Language preprocessing: Highlighting important parts of a sentence while hiding redundancies

- Tokenization: Splitting text into tokens
- Stemming: Removing redundant suffixes
- Lemmatization: Mapping words to common lemmas
- Stopword removal: Removing non-informative words
- Integer encoding: Turning words into numbers

Bag of words

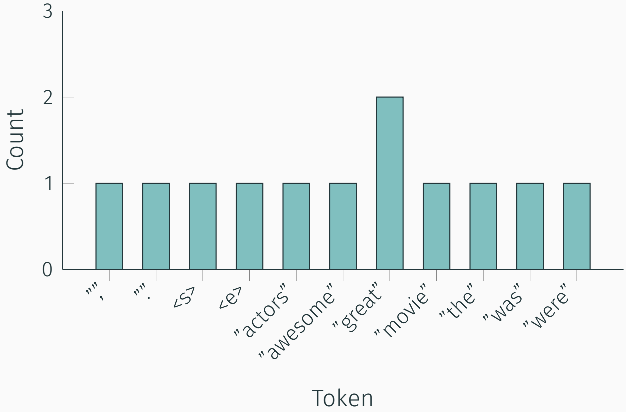
UNIVERSITETET
I OSLO

Bag of words

The movie was great, the actors were awesome.

Bag of words

The movie was great, the actors were awesome.



Bag of words

The movie was great, the actors were awesome.

,	.	<s>	<e>	actors	awesome	great	movie	the	was	were
1	1	1	1	1	1	2	1	1	1	1

Bag of words

The movie was great, the actors were awesome.

,	.	<s>	<e>	actors	awesome	awful	great	horrible	movie	the	was	were	sentiment
1	1	1	1	1	1	0	1	0	1	2	1	1	positive
1	1	1	1	1	0	1	0	1	1	2	1	1	negative

The movie was awful, the actors were horrible.

Bag of words

The movie was great, the actors were awesome.

,	.	<s>	<e>	actors	awesome	awful	great	horrible	movie	the	was	were	sentiment
1	1	1	1	1	1	0	1	0	1	2	1	1	positive
1	1	1	1	1	0	1	0	1	1	2	1	1	negative

The movie was awful, the actors were horrible.

Bag of words

,	.	<s>	<e>	actors	awesome	awful	great	horrible	movie	the	was	were	sentiment
1	1	1	1	1	1	0	1	0	1	2	1	1	positive
1	1	1	1	1	0	1	0	1	1	2	1	1	negative

$$y = \beta_0 + \sum_i \beta_i X_i$$

Bag of words

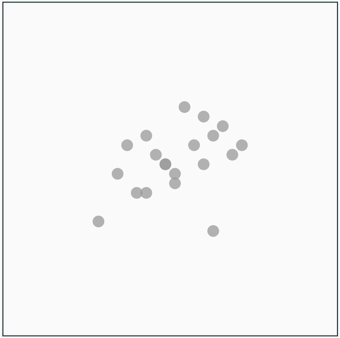
http://localhost:8888/notebooks/notebooks/Bag%20of%20words%20demo.ipynb

Bag of words

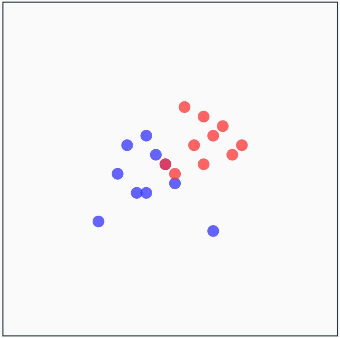
Bag of words: Model language by using word counts (or frequencies)

- Main advantage: Simple, useful when a few key words are sufficient to determine the correct prediction
- Main disadvantage: Does not understand word similarities

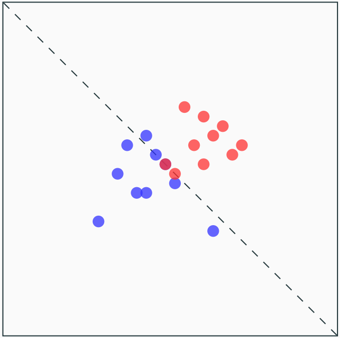
Bag of words: Disadvantages



Bag of words: Disadvantages



Bag of words: Disadvantages



Bag of words: Disadvantages

Dataset: ["This is awesome", "This is wonderful"]

Bag of words: Disadvantages

Dataset: ["This is awesome", "This is wonderful"]

Tokens: [["this" "is" "awesome"], ["this" "is" "wonderful"]]

Bag of words: Disadvantages

Dataset: [”This is awesome”, ”This is wonderful”]

Tokens: [[”this” ”is” ”awesome”], [”this” ”is” ”wonderful”]]

Pruned: [[”awesome”], [”wonderful”]]

Bag of words: Disadvantages

Dataset: ["This is awesome", "This is wonderful"]

Tokens: [["this" "is" "awesome"], ["this" "is" "wonderful"]]

Pruned: [["awesome"], ["wonderful"]]

Dictionary: ["awesome", "wonderful"]

Bag of words: Disadvantages

Dataset: ["This is awesome", "This is wonderful"]

Tokens: [["this" "is" "awesome"], ["this" "is" "wonderful"]]

Pruned: [["awesome"], ["wonderful"]]

Dictionary: ["awesome", "wonderful"]

Encoded:

awesome	wonderful
1	0
0	1

Bag of words: Disadvantages

Dataset: ["This is awesome", "This is wonderful"]

Tokens: [["this" "is" "awesome"], ["this" "is" "wonderful"]]

Pruned: [["awesome"], ["wonderful"]]

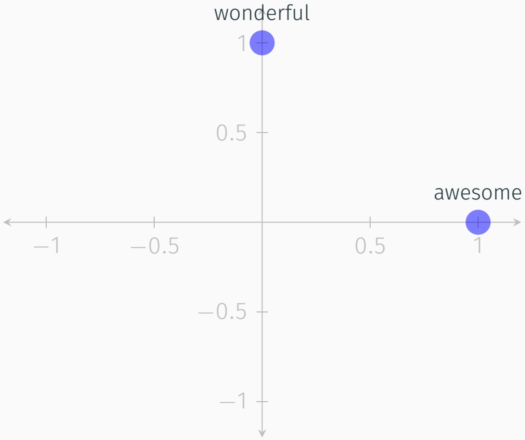
Dictionary: ["awesome", "wonderful"]

Encoded:

awesome	wonderful
1	0
0	1

Vectors: [[1, 0], [0, 1]]

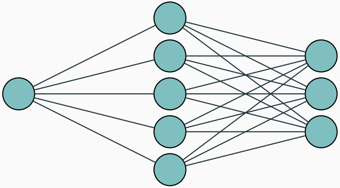
Bag of words: Disadvantages



Embeddings

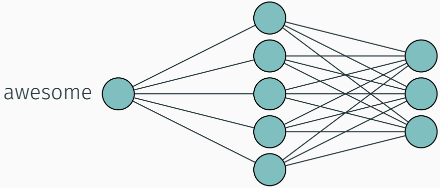
Word2vec

The movie was awesome.
The movie was wonderful.
The movie was fantastic.



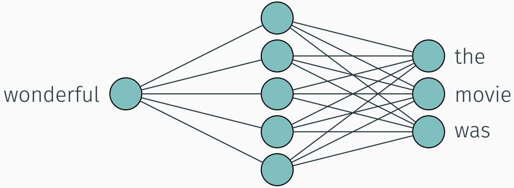
Word2vec

The movie was awesome.
The movie was wonderful.
The movie was fantastic.



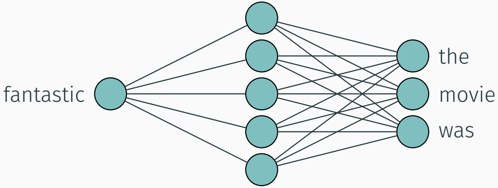
Word2vec

The movie was awesome.
The movie was wonderful.
The movie was fantastic.



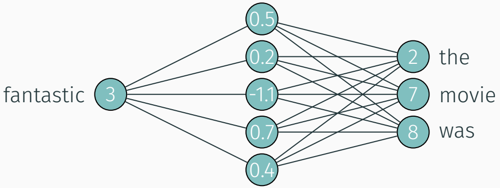
Word2vec

The movie was awesome.
The movie was wonderful.
The movie was fantastic.



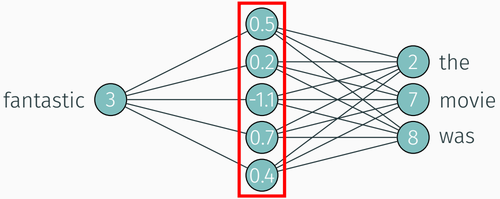
Word2vec

The movie was awesome.
The movie was wonderful.
The movie was fantastic.



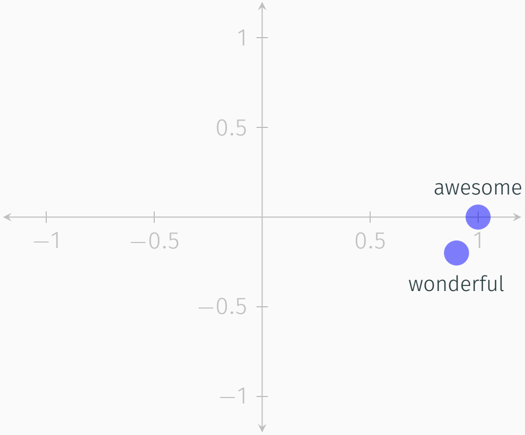
Word2vec

The movie was awesome.
The movie was wonderful.
The movie was fantastic.



fantastic=[0.5, 0.2, -1.1, 0.7, 0.4]

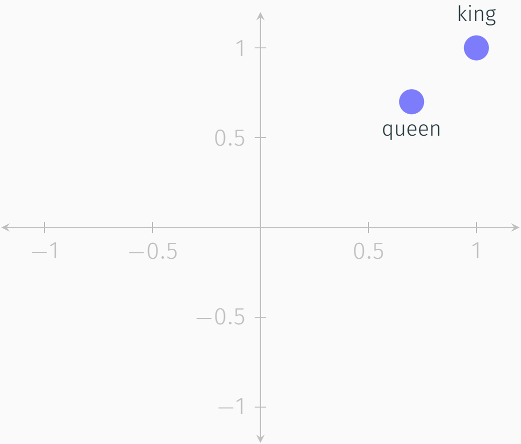
Word2vec



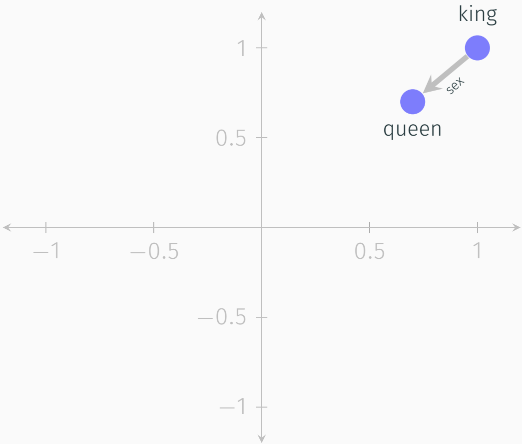
Word2vec

The movie was awesome.
The food was awesome.
The book was awesome.

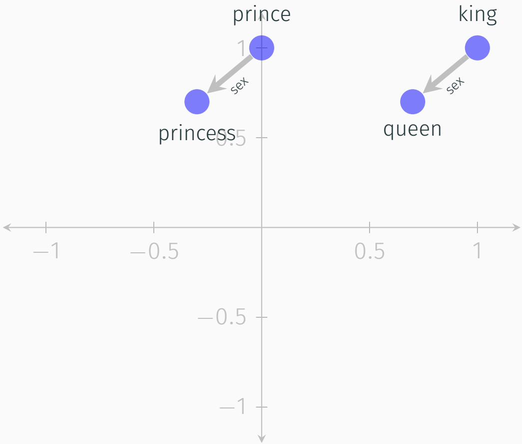
Word2vec



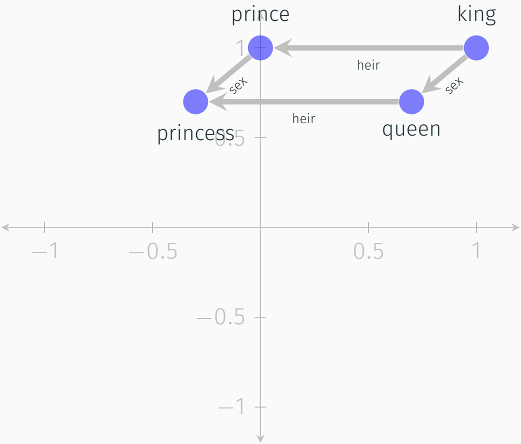
Word2vec



Word2vec



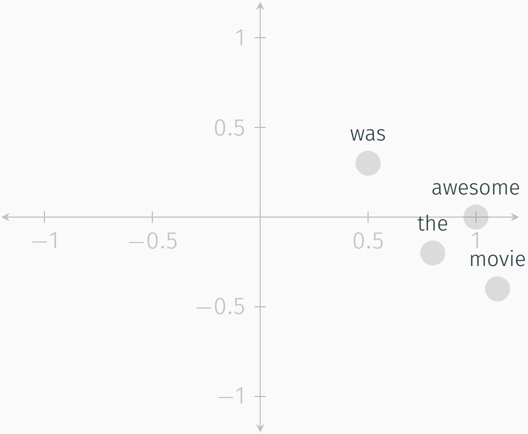
Word2vec



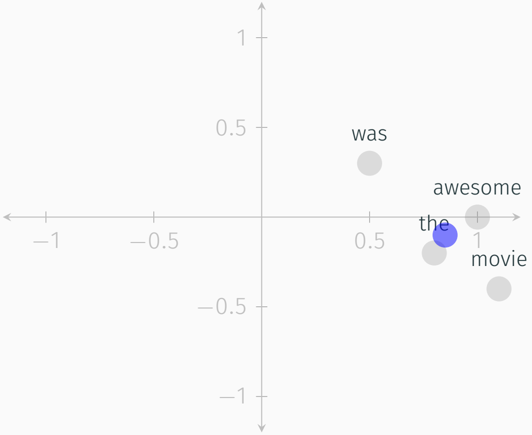
Word2vec

word2vec(queen)= word2vec(king)
-word2vec(man)
+word2vec(woman)

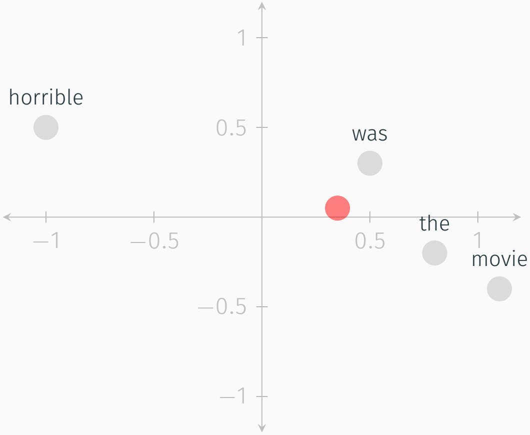
Word2vec



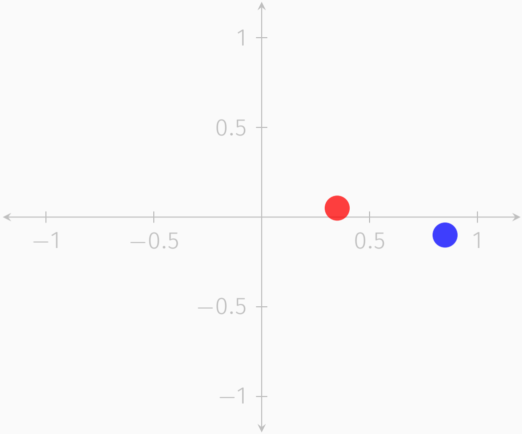
Word2vec



Word2vec



Word2vec



Word2vec

http://localhost:8888/notebooks/notebooks/Word2vec%20demo.ipynb

Word2vec: Disadvantages

I think the movie was really bad, but my friend said it was good.

=

I think the movie was really good, but my friend said it was bad.

Word2vec

Word2vec: Model words by vectors that encode their semantic content

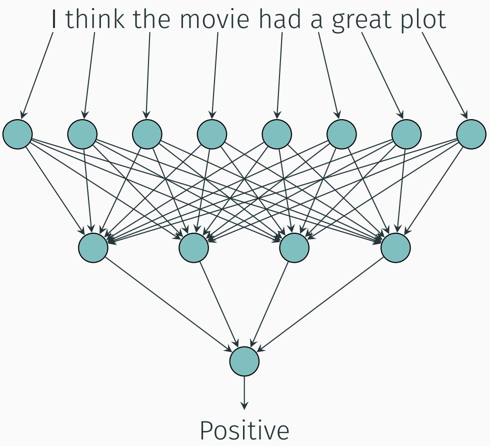
- Main advantage: Models semantic meaning, allowing us to do mathematics with language
- Main disadvantage: Does not consider the structure innate to language

Recurrent neural networks

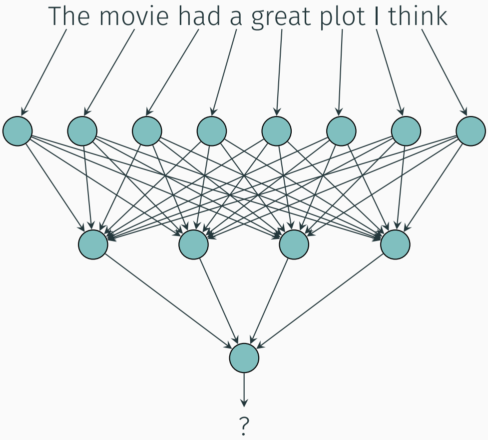
Recurrent neural networks

I think the movie had a great plot

Recurrent neural networks



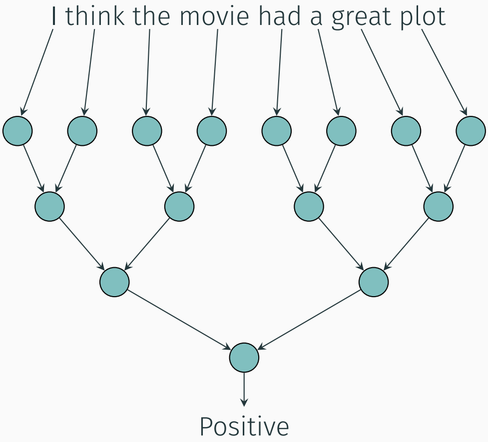
Recurrent neural networks



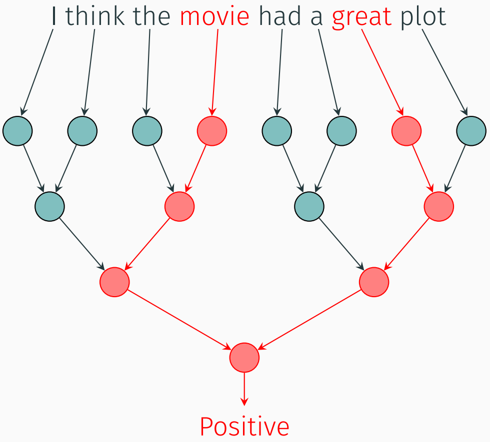
Recurrent neural networks

I think the movie had a great plot

Recurrent neural networks



Recurrent neural networks



Recurrent neural networks

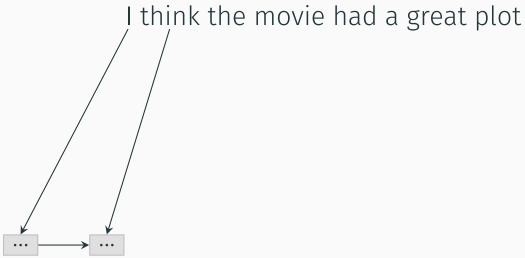
I think the movie had a great plot

Recurrent neural networks

I think the movie had a great plot

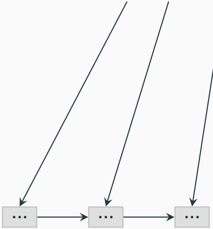
...

Recurrent neural networks

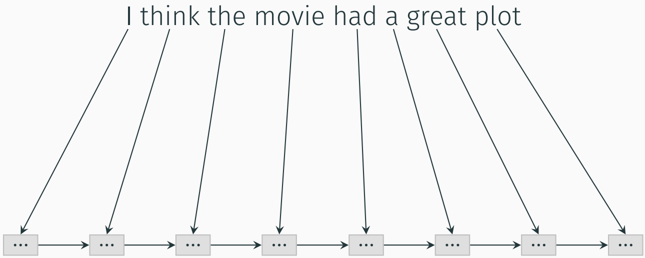


Recurrent neural networks

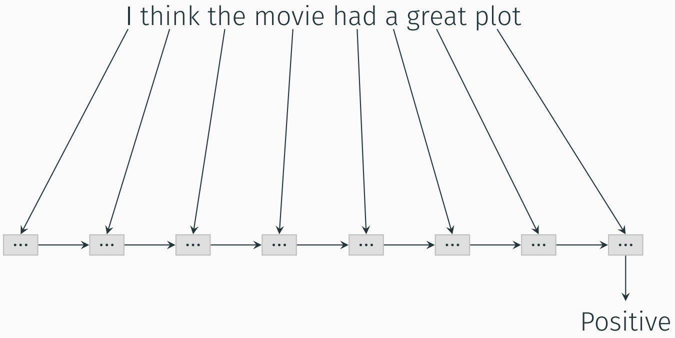
I think the movie had a great plot



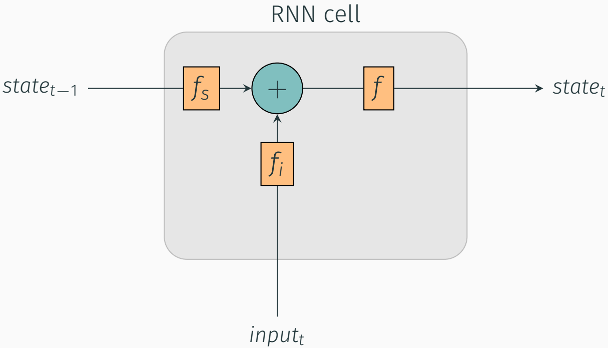
Recurrent neural networks



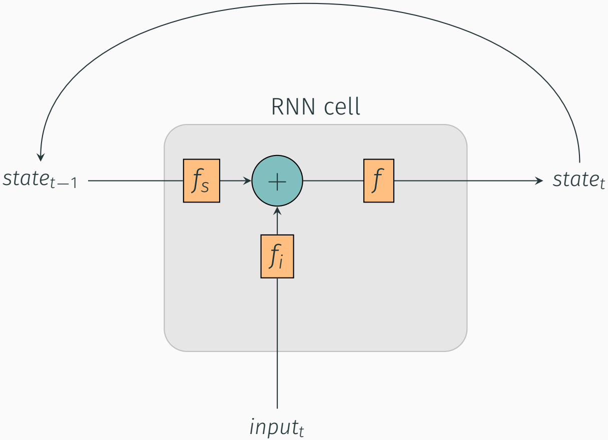
Recurrent neural networks



Recurrent neural networks

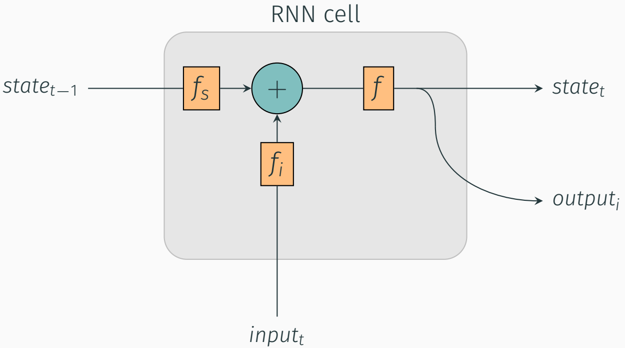


Recurrent neural networks



Recurrent neural networks

Blackboard demo!


RNN cell
$state_{t-1}$
f_s
$+$
f
$state_t$
f_i
$output_i$
$input_t$

Recurrent neural networks

More blackboard demo!

LSTM Cell

Recurrent neural networks

https://colab.research.google.com/drive/1MHTzUMViR8vKGOI-VbZR0CYzsJMkSrJw

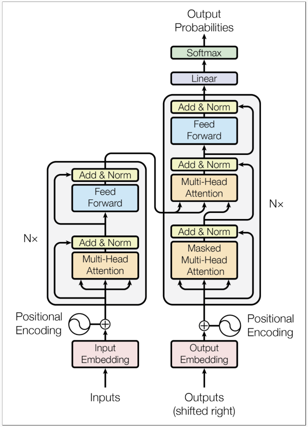
Recurrent neural networks

RNNs: Models sequences by recursively considering what it has seen so far, and what the new input token is

- Main advantage: Is able to encompass both long- and short-term dependencies
- Main disadvantage: In practice it is hard to weigh long-term versus short-term

Transformers

Transformers



Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30

Transformers: Attention

The movie was great, the actors were________

Transformers: Attention

The movie was great, the actors were________

Transformers: Attention

The movie was great, the actors were ________

Transformers: Attention

The movie was great, the actors were ________

Transformers: Attention

The movie was great, the actors were ________

Transformers: Attention

The movie was great, the actors were ________

Transformers: Attention

The movie was great, the actors were ________

Transformers: Attention

The movie was great, the actors were________

Transformers: Attention

The movie was great, the actors were________

Transformers: Attention

The movie was great, we saw it at the new Cinema in the city center, right down by the restaurant where we went for my birthday that one year, the one where the clown was inside the cake, the actors were ________

Transformers: Attention

The movie was great, we saw it at the new Cinema in the city center, right down by the restaurant where we went for my birthday that one year, the one where the clown was inside the cake, the actors were ________

Transformers: Attention

The movie was great, we saw it at the new Cinema in the city center, right down by the restaurant where we went for my birthday that one year, the one where the clown was inside the cake, the actors were ________

Transformers: Attention

The movie was great, the actors were ________

Transformers: Attention

The movie was great, the actors were ________

$[8 \quad 7 \quad 9 \quad 6 \; 0 \; 8 \quad 4 \quad 10] \rightarrow ?$

Transformers: Attention

The movie was great, the actors were ________

$$[8 \quad 7 \quad 9 \quad 6 \; 0 \; 8 \quad 4 \quad 10] \rightarrow ?$$

$$[0 \quad 0 \quad 0 \quad 1 \; 0 \; 0 \quad 0 \quad 0]$$

Transformers: Attention

The movie was great, the actors were ________

$$
\begin{array}{cccccccccc}
[8 & 7 & 9 & 6 & 0 & 8 & 4 & 10] & \rightarrow & ? \\
 & & & \times & & & & & & \\
[0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\,] & & \\
 & & & = & & & & & & \\
[0 & 0 & 0 & 6 & 0 & 0 & 0 & 0\,] & &
\end{array}
$$

Transformers: Positional encoding

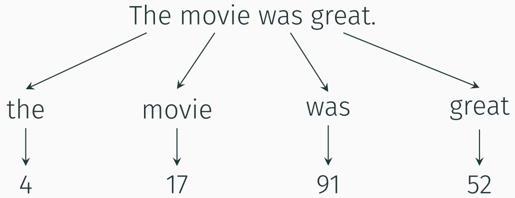
The movie was great.

Transformers: Positional encoding

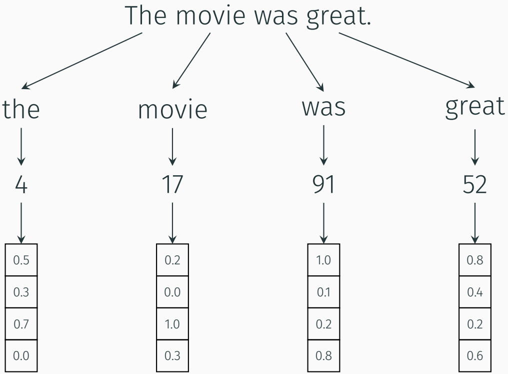
The movie was great.

the movie was great

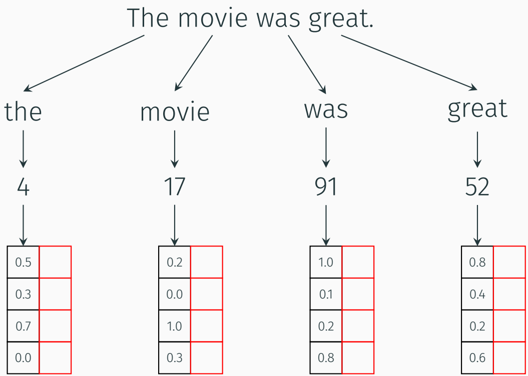
Transformers: Positional encoding



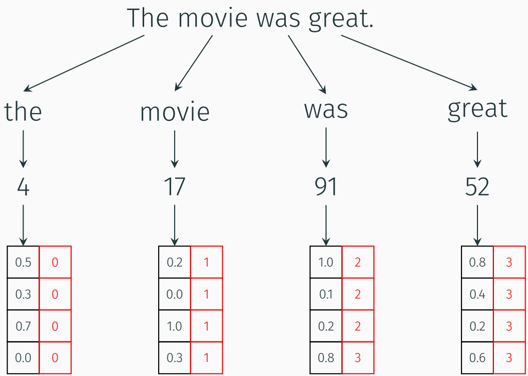
Transformers: Positional encoding

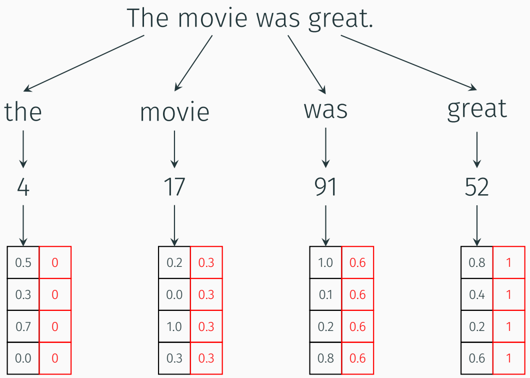


Transformers: Positional encoding

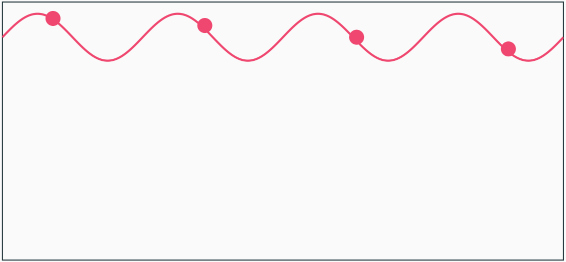


Transformers: Positional encoding



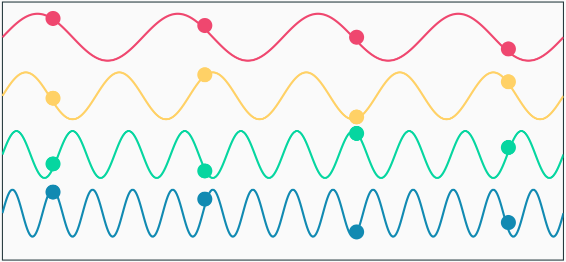

The movie was great.
the
movie
was
great
4
17
91
52
0.5 0
0.3 0
0.7 0
0.0 0
0.2 0.3
0.0 0.3
1.0 0.3
0.3 0.3
1.0 0.6
0.1 0.6
0.2 0.6
0.8 0.6
0.8 1
0.4 1
0.2 1
0.6 1

Transformers: Positional encoding



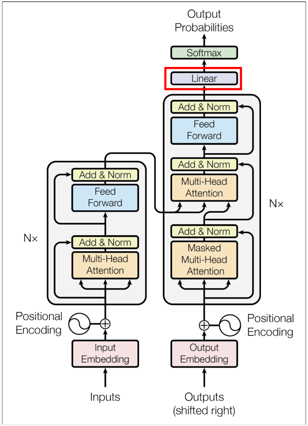
0.9	0.3	-0.3	-0.9
0.9	0.3	-0.3	-0.9
0.9	0.3	-0.3	-0.9
0.9	0.3	-0.3	-0.9

Transformers: Positional encoding



0.9	0.3	-0.3	-0.9
0.5	0.9	-0.9	0.8
-0.3	-0.9	-0.9	-0.5
1.0	0.8	-0.9	-0.6

Transformers: Embedding



Transformers: Demo

https://huggingface.co/docs/transformers/model_doc/llama2

Transformers: Demo

http://localhost:8888/notebooks/notebooks/GPT%20Embedding%20demo.ipynb

Transformers

Transformers: Revolutionized language modelling by combining feed forward neural networks with multihead attention and positional endcodings (and infinite data and compute)

- Main advantage: Outperforms everything else for almost all language modelling tasks
- Main disadvantage: Can either be used locally, which is fidgety and requires a good computer, or via an API, which is costly and gives others access to your data