Введение в нейронные сети

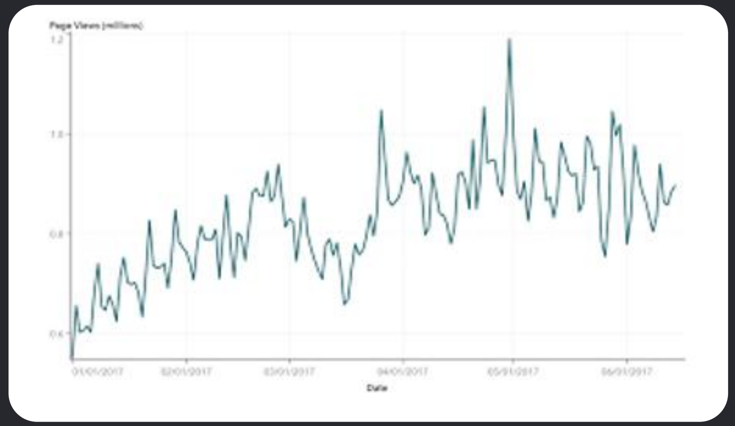
Лекция 3. Введение в рекуррентные НС

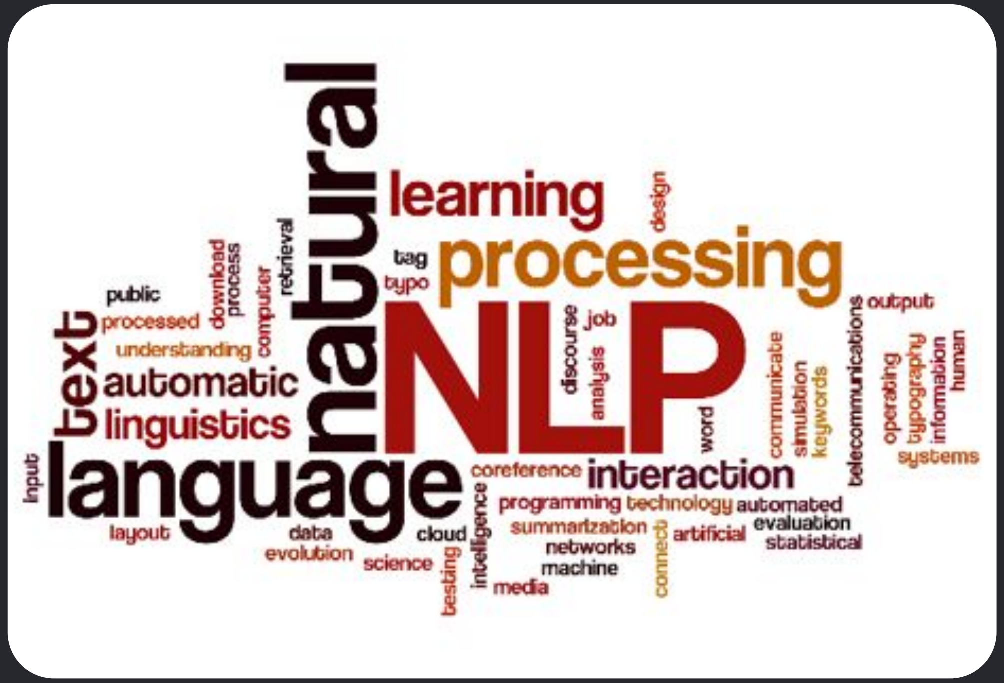
Оглавление

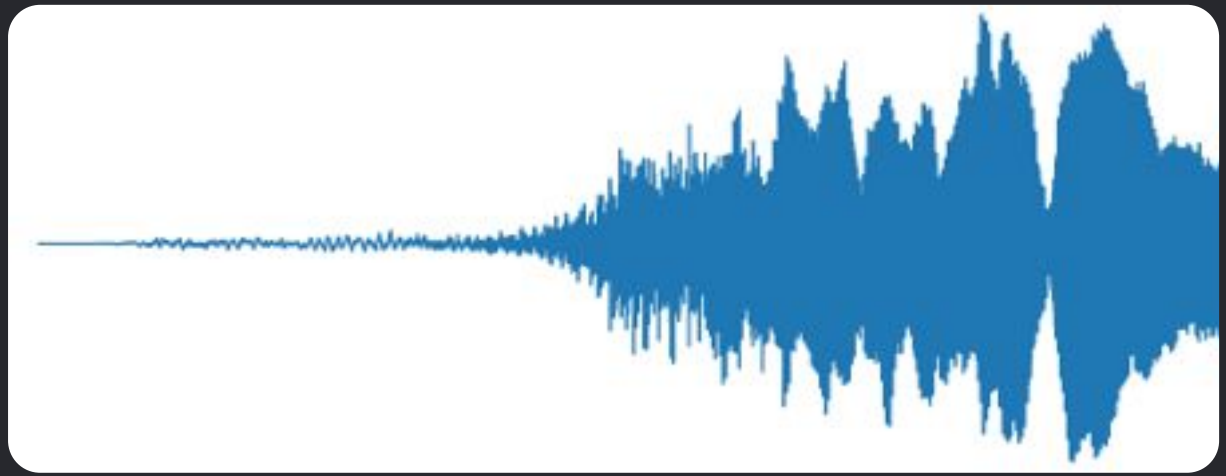
Хотел сделать как лучше, а получилось про работу с последовательностями

Типы данных (модальности)

Owner	Country	File_Date	IPC_Class
Company A	US	6/18/2008	H05H13
Company A	EP	1/30/1998	A61N5
Company A	EP	1/30/1998	A61N5
Company A	EP	1/30/1998	A61N5
Company A	JP	8/28/1997	A61N5
Company A	JP	10/4/2002	A61N5
Company A	JP	1/27/2003	A61N5
Company A	JP	4/14/2003	A61N5
Company A	JP	5/13/2011	A61N5
Company B	JP	4/2/1998	G12B13
Company B	JP	4/2/1998	G12B13
Company B	JP	5/28/1997	A61N5
Company B	JP	11/12/1997	A61N5
Company B	JP	2/29/2000	A61N5
Company B	JP	4/30/2002	A61N5





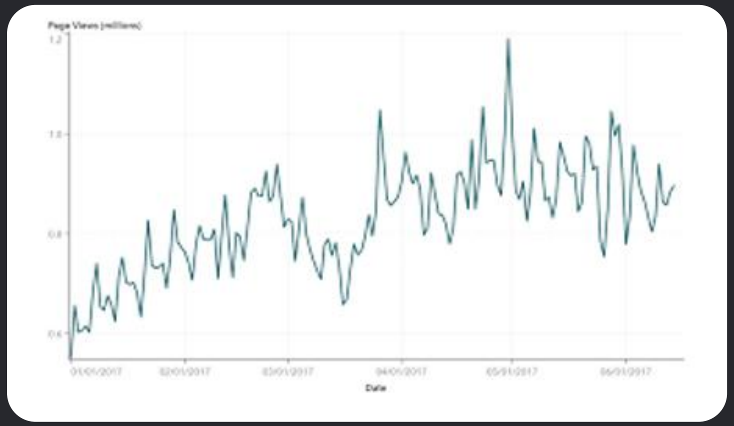


Типы данных (модальности)

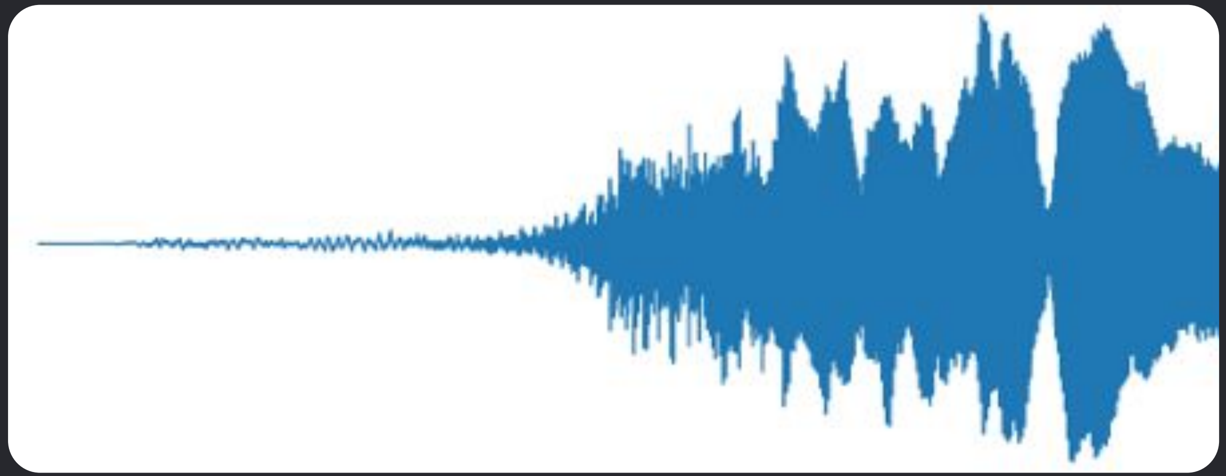
Owner	Country	File_Date	IPC_Class
Company A	US	6/18/2008	H05H13
Company A	EP	1/30/1998	A61N5
Company A	EP	1/30/1998	A61N5
Company A	EP	1/30/1998	A61N5
Company A	JP	8/28/1997	A61N5
Company A	JP	10/4/2002	A61N5
Company A	JP	1/27/2003	A61N5
Company A	JP	4/14/2003	A61N5
Company A	JP	5/13/2011	A61N5
Company B	JP	4/2/1998	G12B13
Company B	JP	4/2/1998	G12B13
Company B	JP	5/28/1997	A61N5
Company B	JP	11/12/1997	A61N5
Company B	JP	2/29/2000	A61N5
Company B	JP	4/30/2002	A61N5

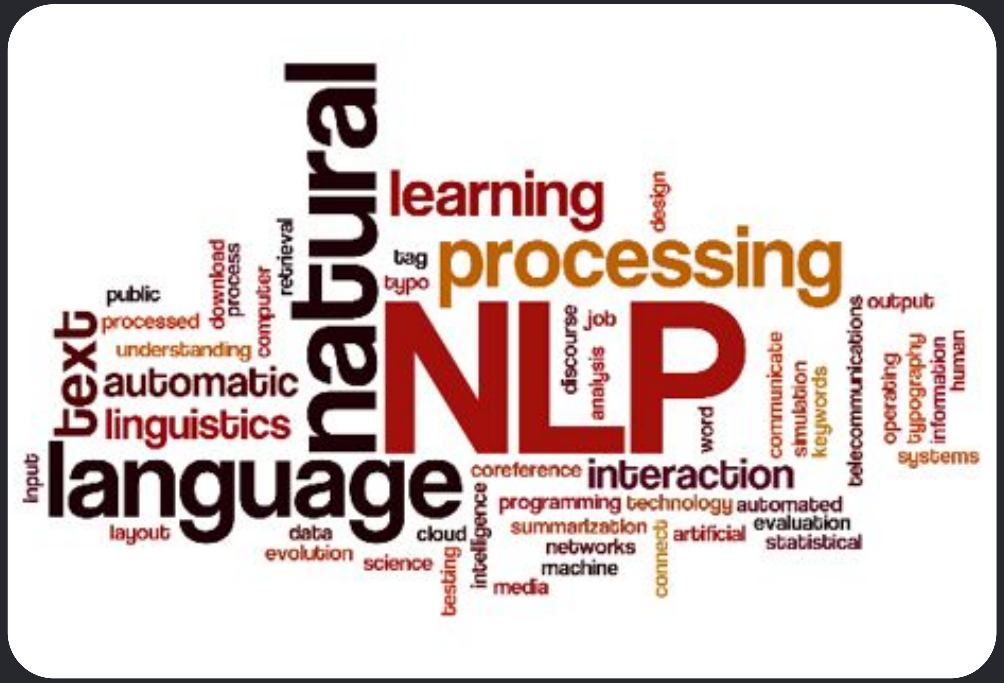
Tabular

Images



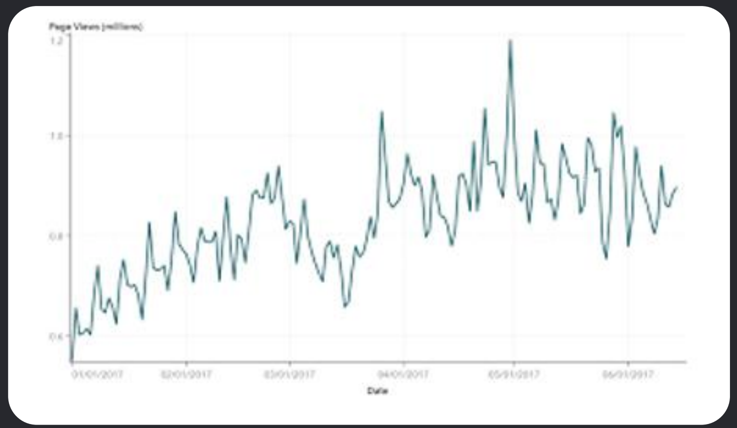
Sequences

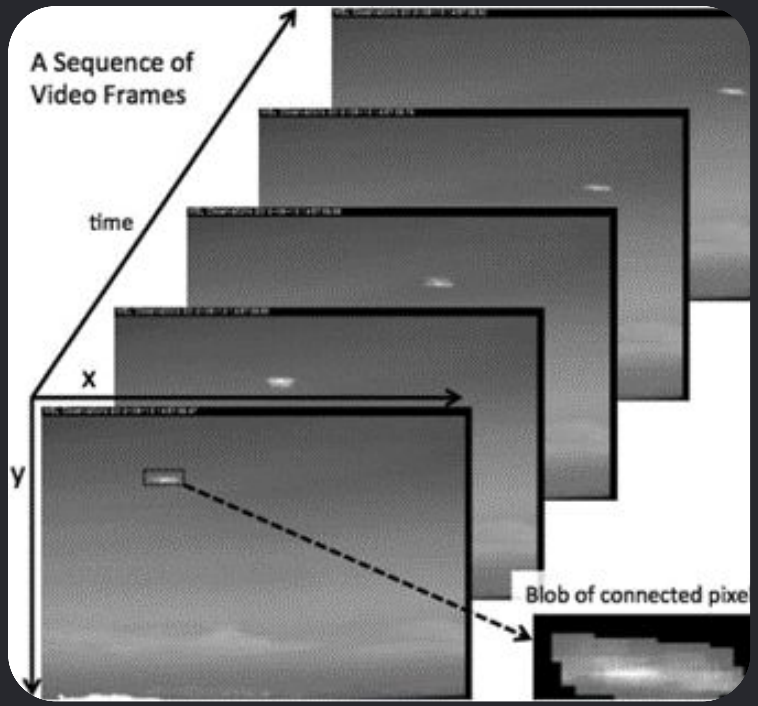


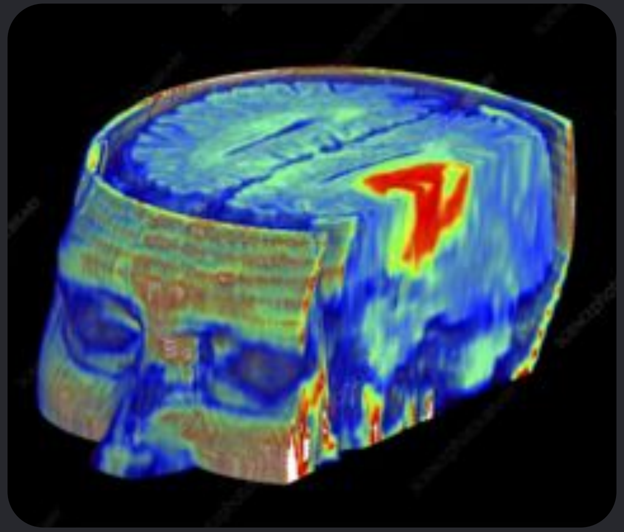


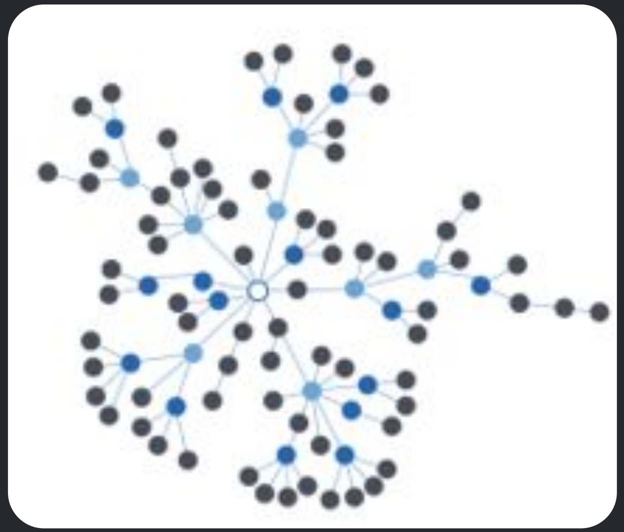
Типы данных (модальности)

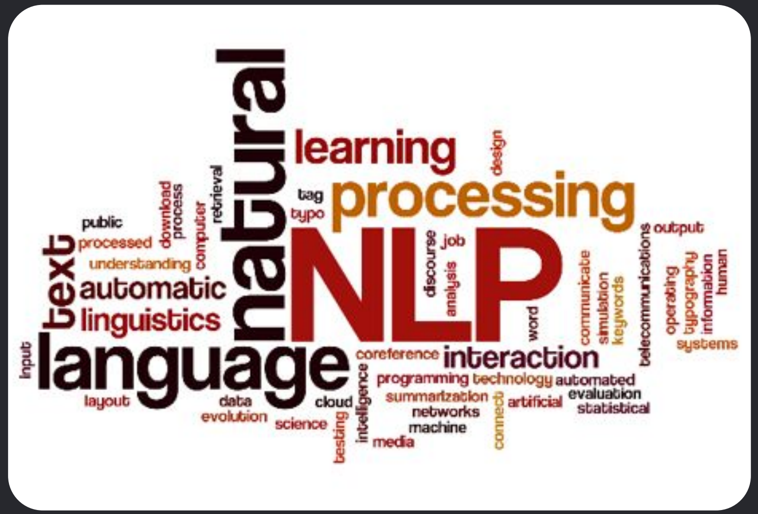
Owner	Country	File_Date	IPC_Class
Company A	US	6/18/2008	H05H13
Company A	EP	1/30/1998	A61N5
Company A	EP	1/30/1998	A61N5
Company A	EP	1/30/1998	A61N5
Company A	JP	8/28/1997	A61N5
Company A	JP	10/4/2002	A61N5
Company A	JP	1/27/2003	A61N5
Company A	JP	4/14/2003	A61N5
Company A	JP	5/13/2011	A61N5
Company B	JP	4/2/1998	G12B13
Company B	JP	4/2/1998	G12B13
Company B	JP	5/28/1997	A61N5
Company B	JP	11/12/1997	A61N5
Company B	JP	2/29/2000	A61N5
Company B	JP	4/30/2002	A61N5

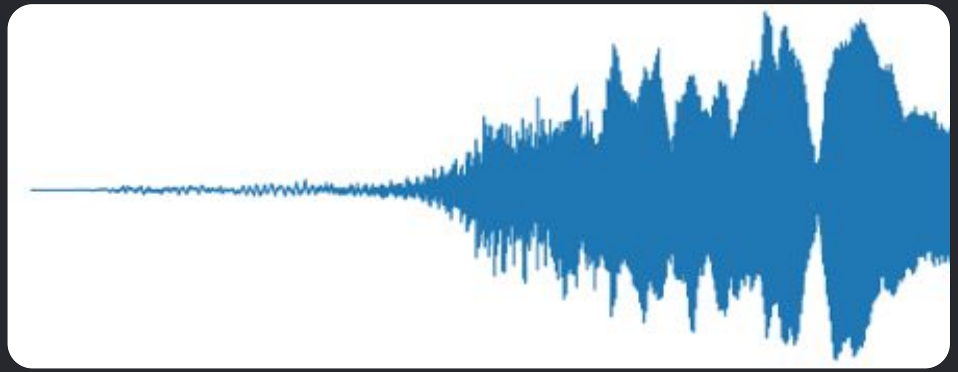












Основная идея DL — преобразовывать данные в тензор

Finally, NLP!

«Finally, a computer that understands you like your mother»

- Наконец-то компьютер, который понимает вас так же хорошо, как ваша мама.
- Наконец-то компьютер, который понимает, что вам нравится ваша мама.
- Наконец-то компьютер, который понимает вас так же хорошо, как он понимает вашу маму.

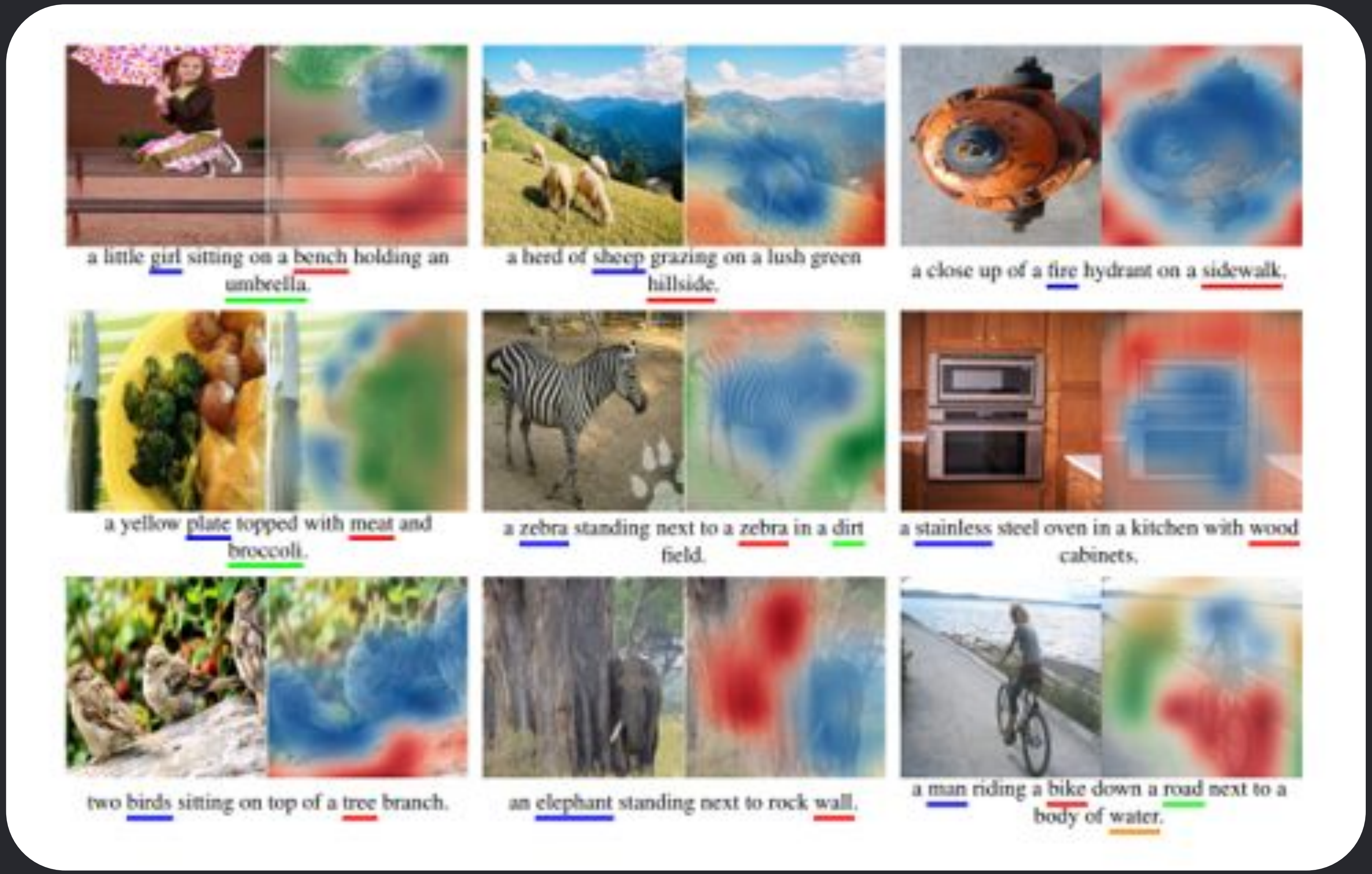
Newspaper headlines

- Boy paralyzed after tumor fights back to gain black belt.
- Miners refuse to work after death.
- The Pope's baby steps on gays.

As English not all languages words in the same order put. Hmmmmmm

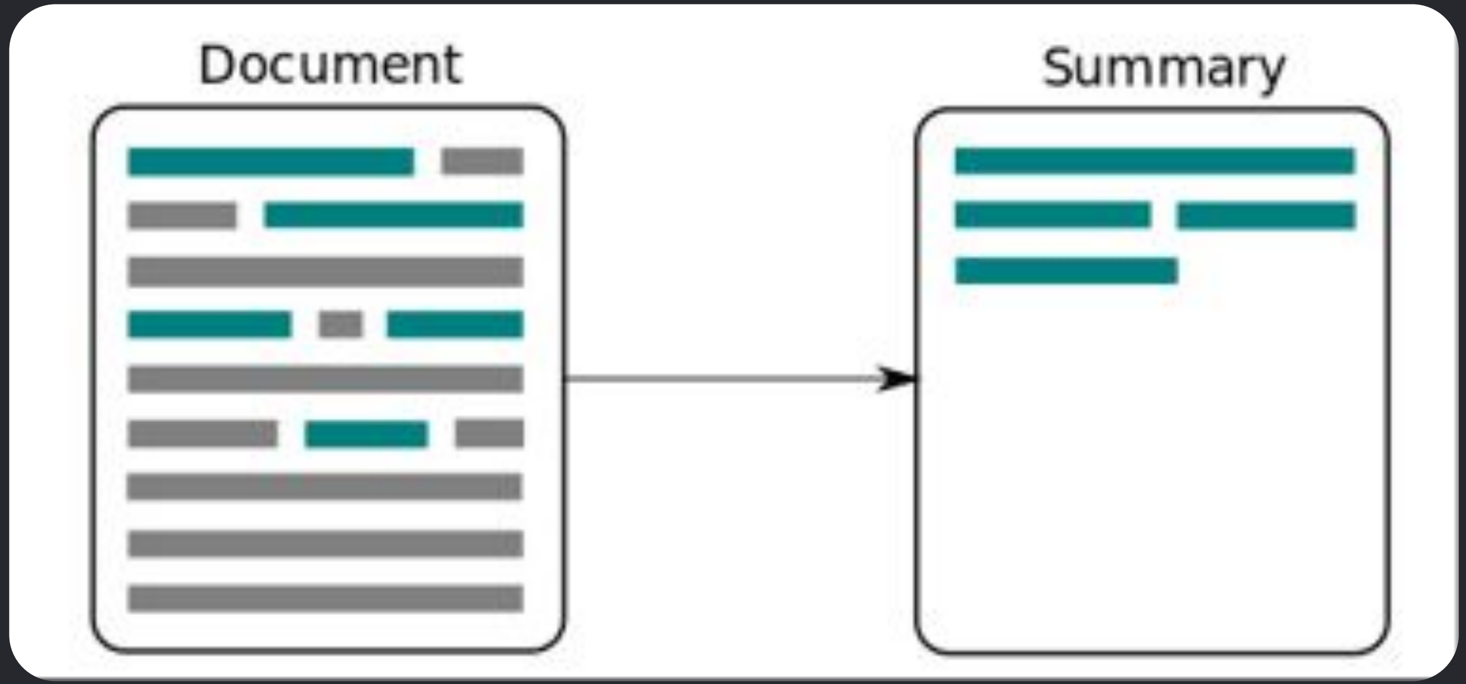
Yoda

- 彼は電車で学校に行ってきました。
- He train by school to went.

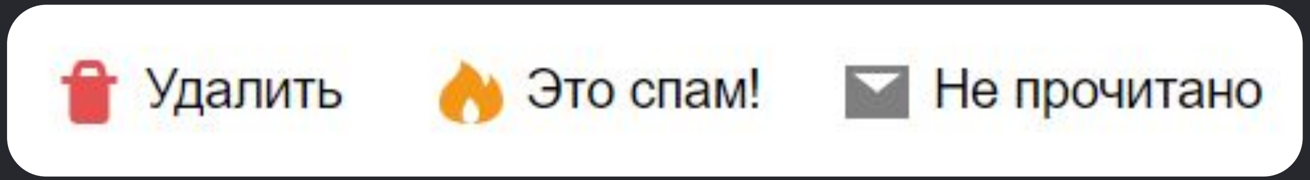

a little girl sitting on a bench holding an umbrella.
a herd of sheep grazing on a lush green hillside.
a close up of a fire hydrant on a sidewalk.
a yellow plate topped with meat and broccoli.
a zebra standing next to a zebra in a dirt field.
a stainless steel oven in a kitchen with wood cabinets.
two birds sitting on top of a tree branch.
an elephant standing next to rock wall.
a man riding a bike down a road next to a body of water.


English
Español
Translate
Italiano


data
techniques
information
language
human
Text
Mining
algorithms
linguistics

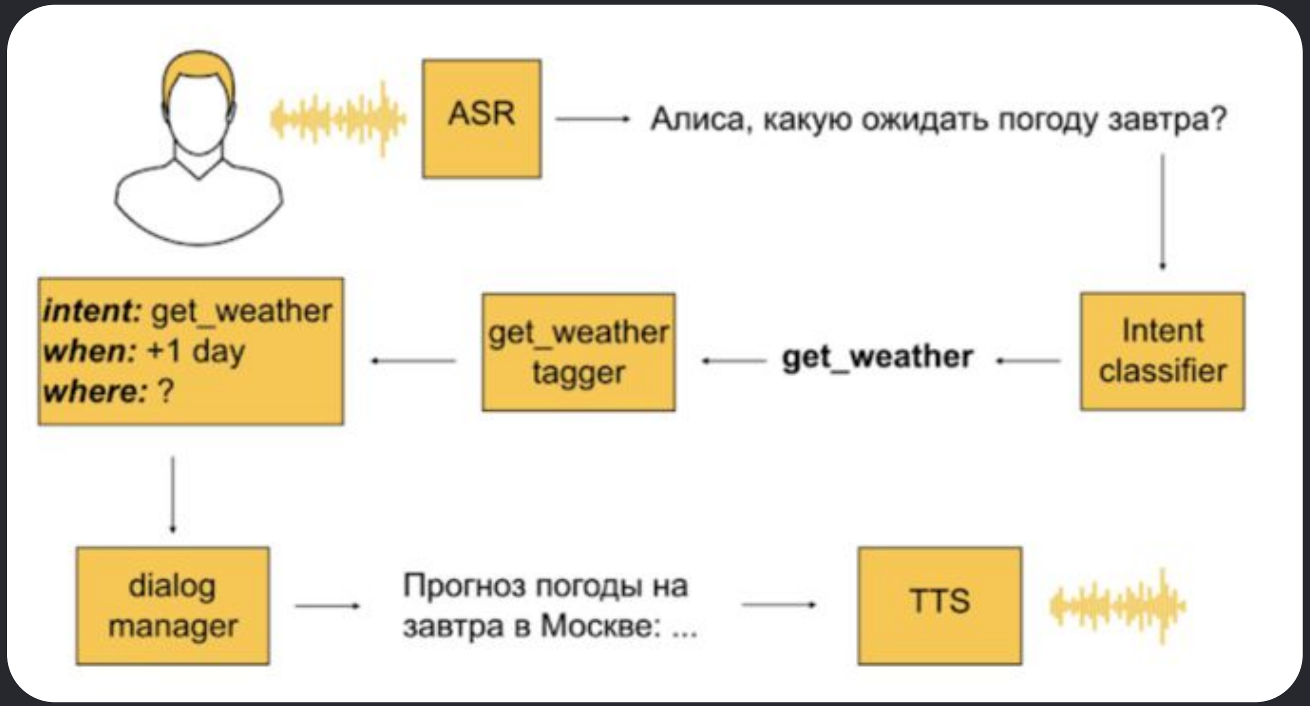

Document
Summary

Классификация текстов



Как самодостаточная задача:

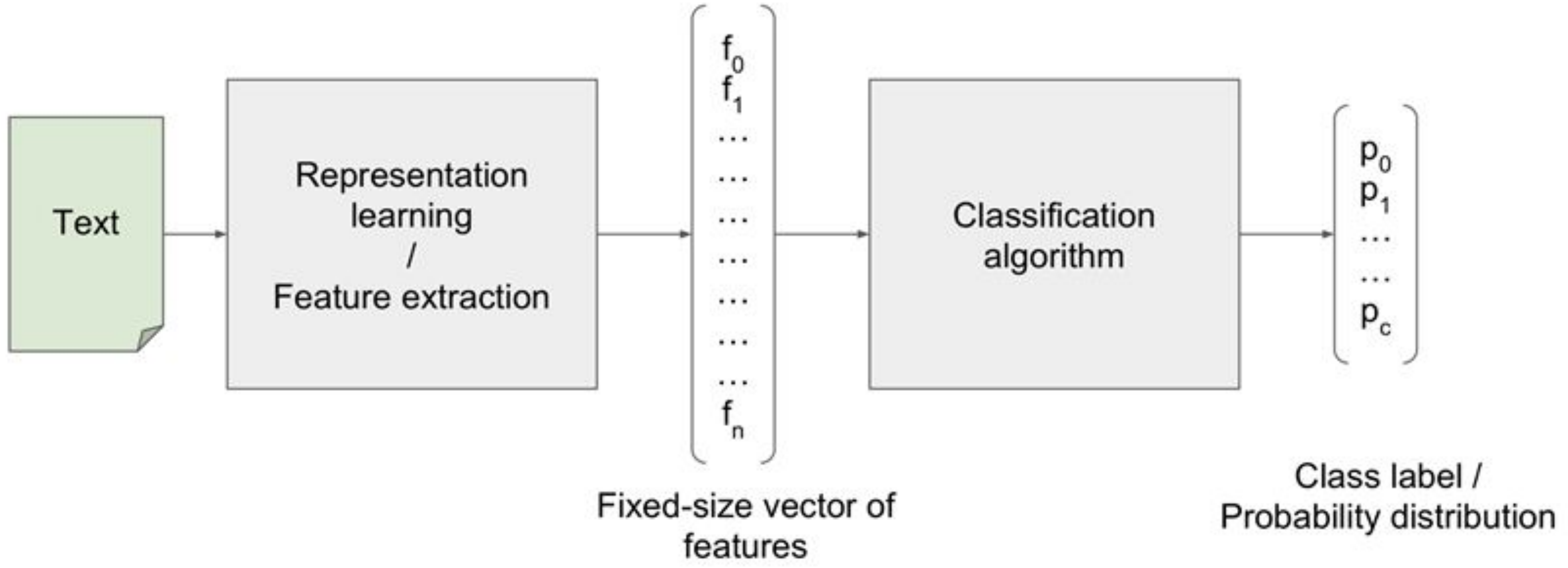
- spam-filtering,
- sentiment analysis,
- fake news or clickbait protection,
- troll or bot protection.



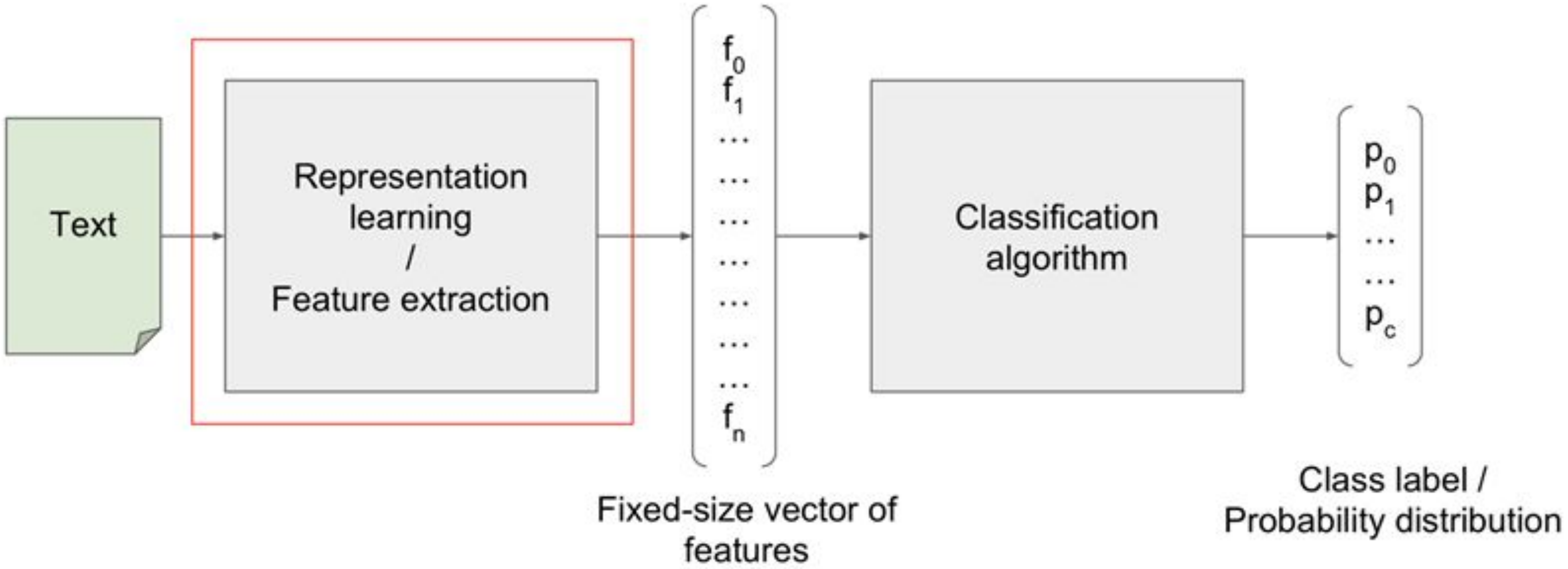
Как часть более сложной системы:

- intent classification in dialogue systems,
- hybrid MT systems.

Text classification in general



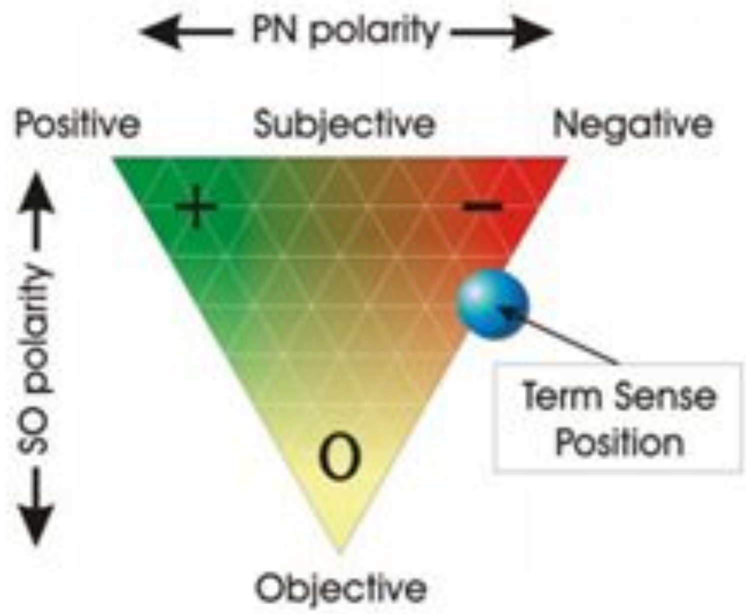
Text classification in general



Text representation: feature engineering

As for many ML tasks, it is possible to generate useful features by hands.

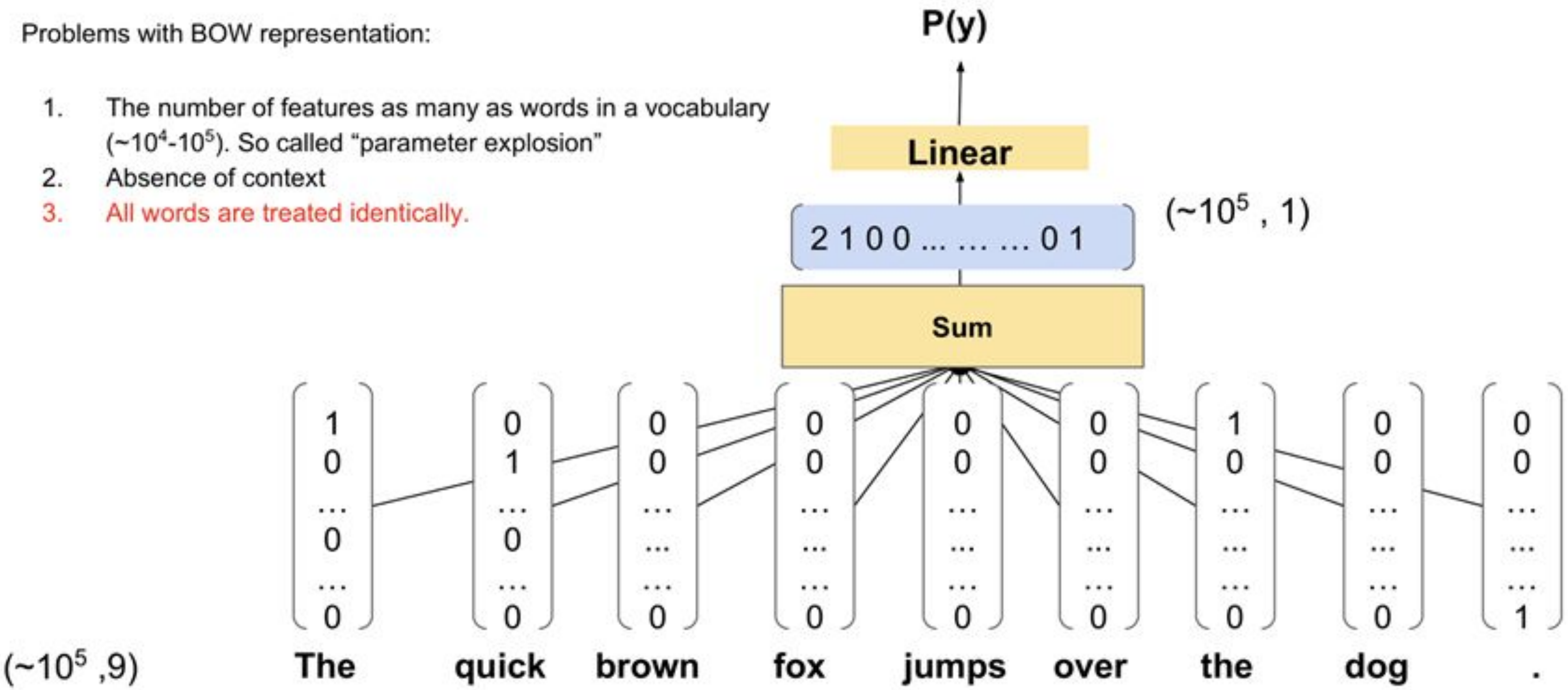
- General statistics: text length, text length variance,...
- Scores from tagged word lists:
  - Sentiment dictionaries: SentiWordNet, SentiWords, ...
  - Subjectivity/objectivity dictionaries: MPQA
  - ...
- Syntactic features:
  - POS tags
- Ad-hoc features: e.g. number of emojis (🤮 or 🤬)



Sparse text representation: BOW

Problems with BOW representation:

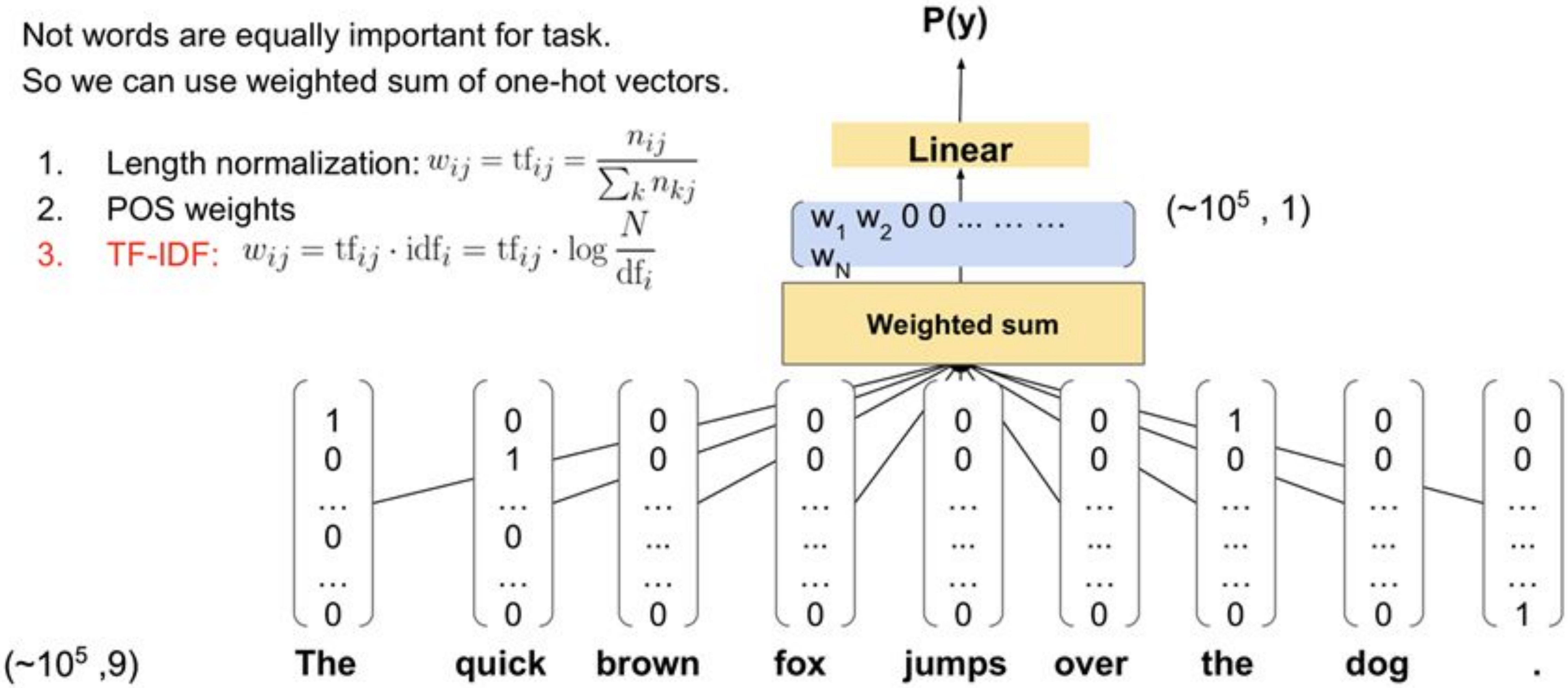
1. The number of features as many as words in a vocabulary ($\sim 10^4$-10^5). So called "parameter explosion"
2. Absence of context
3. All words are treated identically.



Weighting techniques for BOW

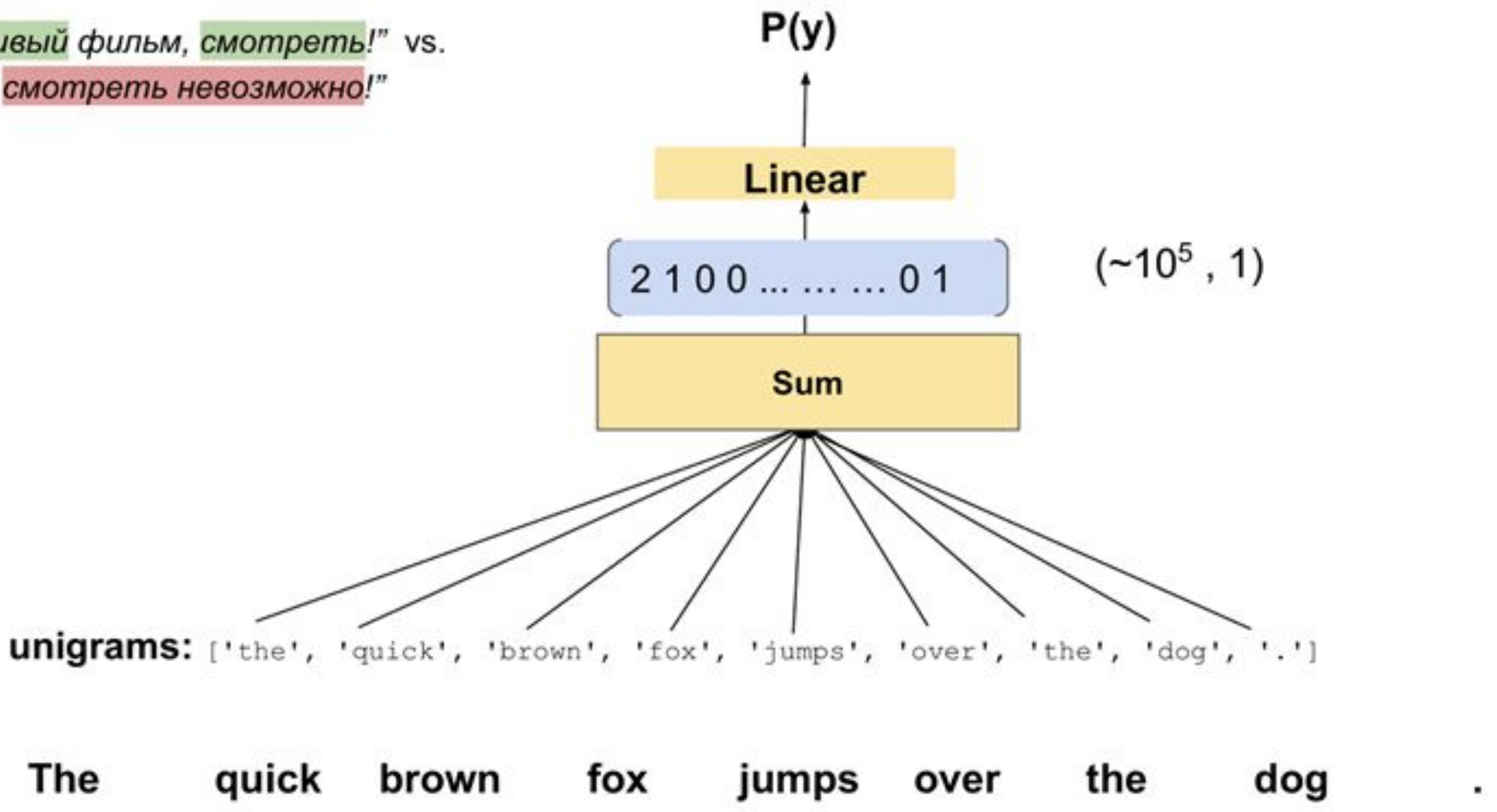
Not words are equally important for task.
So we can use weighted sum of one-hot vectors.

1. Length normalization: $w_{ij} = \mathrm{tf}_{ij} = \frac{n_{ij}}{\sum_k n_{kj}}$
2. POS weights
3. TF-IDF: $w_{ij} = \mathrm{tf}_{ij} \cdot \mathrm{idf}_i = \mathrm{tf}_{ij} \cdot \log \frac{N}{\mathrm{df}_i}$



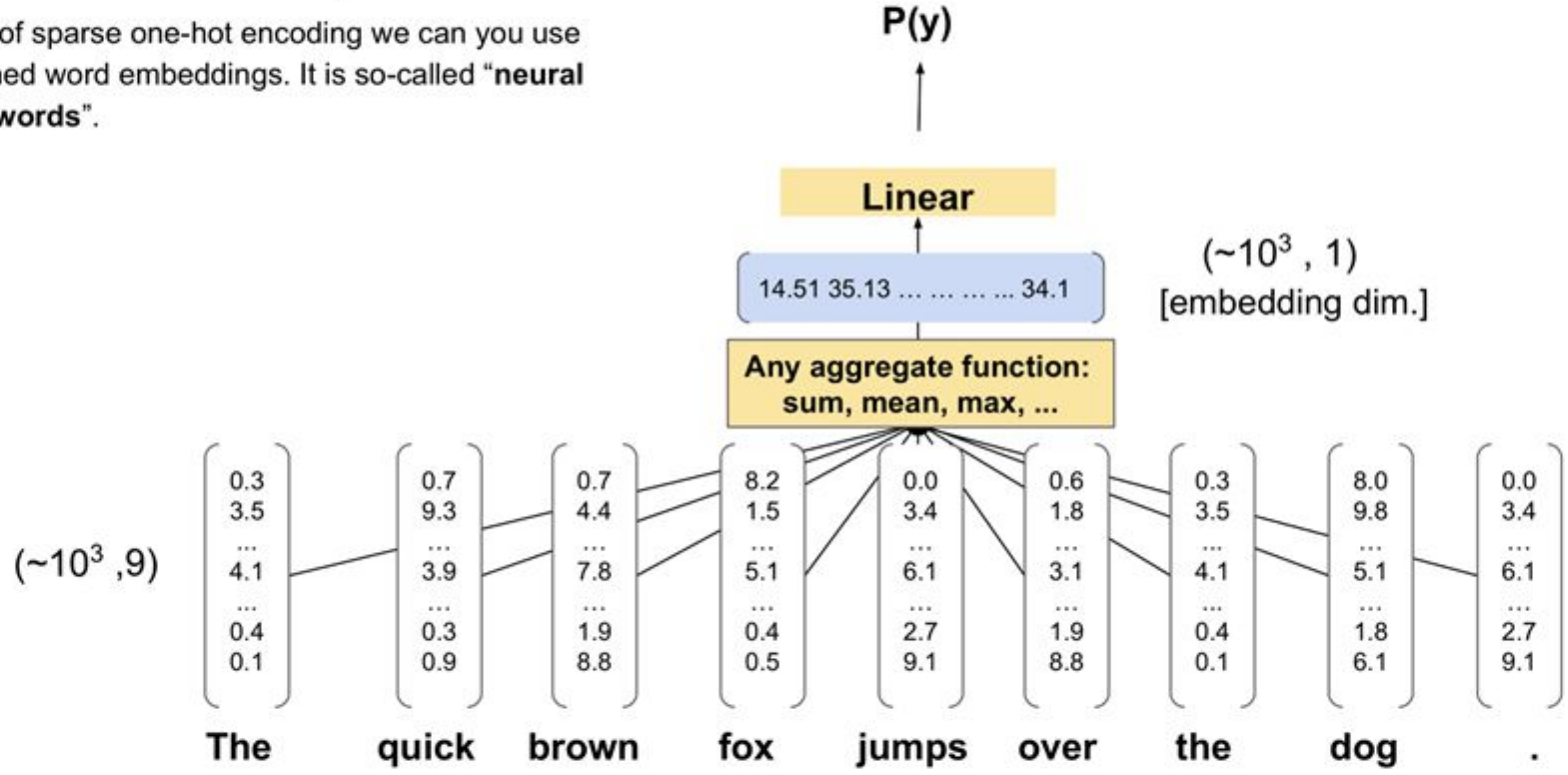
Context importance

"Невозможно красивый фильм, смотреть!" vs.
"Красивый фильм, смотреть невозможно!"



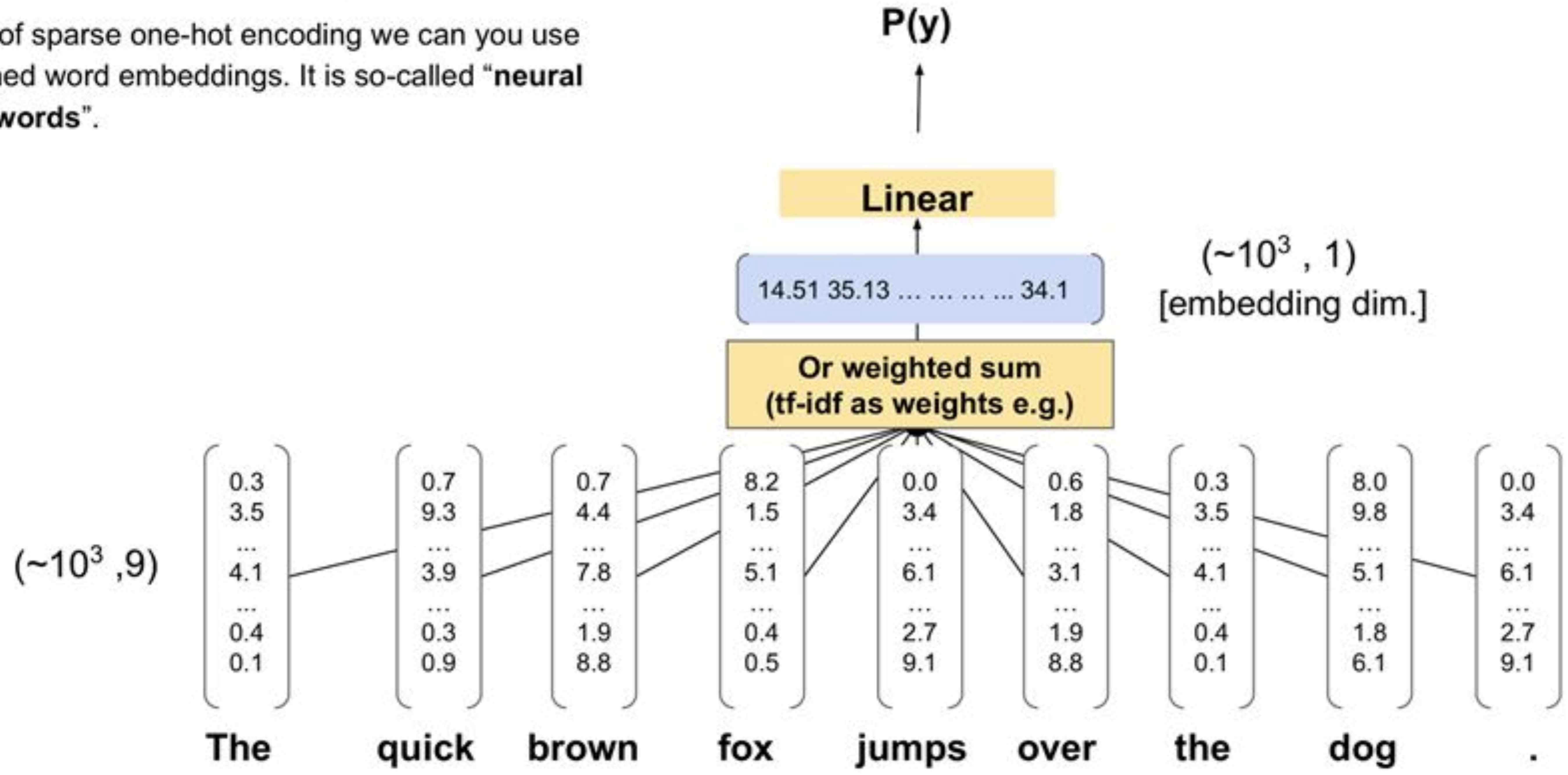
Dense text representation: NBOW

Instead of sparse one-hot encoding we can you use pre-trained word embeddings. It is so-called "**neural bag-of-words**".



Dense text representation: NBOW

Instead of sparse one-hot encoding we can you use pre-trained word embeddings. It is so-called “**neural bag-of-words**”.



BOW and NBOW: the shared problems

1. The importance weights for the word vectors aren’t defined fully.
2. The only way to use context for these models is to utilize word ngrams.

🤔 Hmm...

BOW and NBOW: the shared problems

1. The importance weights for the word vectors aren’t defined fully.
2. The only way to use context for these models is to utilize word ngrams.

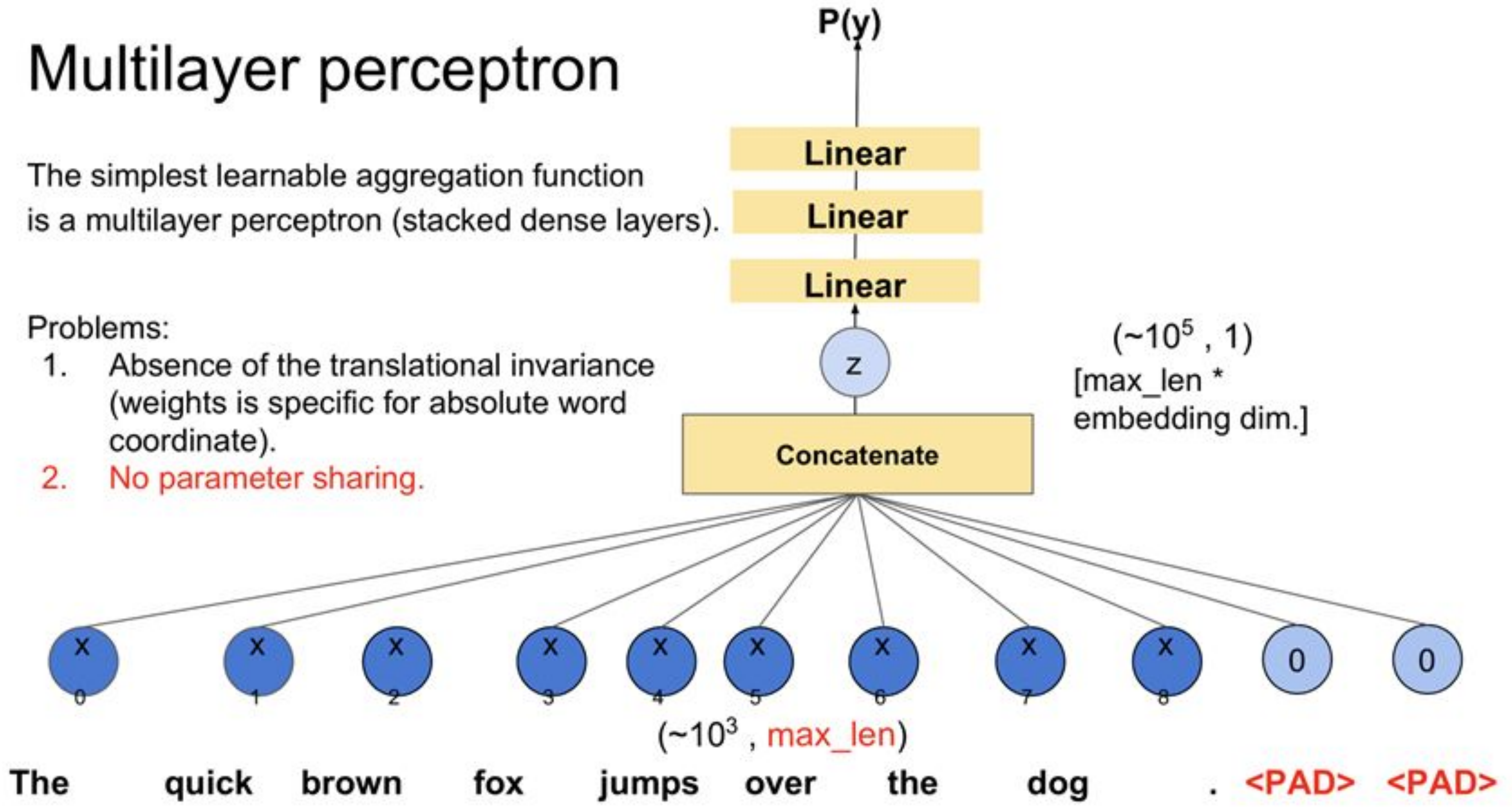
We can use a learnable aggregation function to overcome the difficulties.
The learnable function is a neural network (the universal approximator)

Multilayer perceptron

The simplest learnable aggregation function
is a multilayer perceptron (stacked dense layers).

Problems:

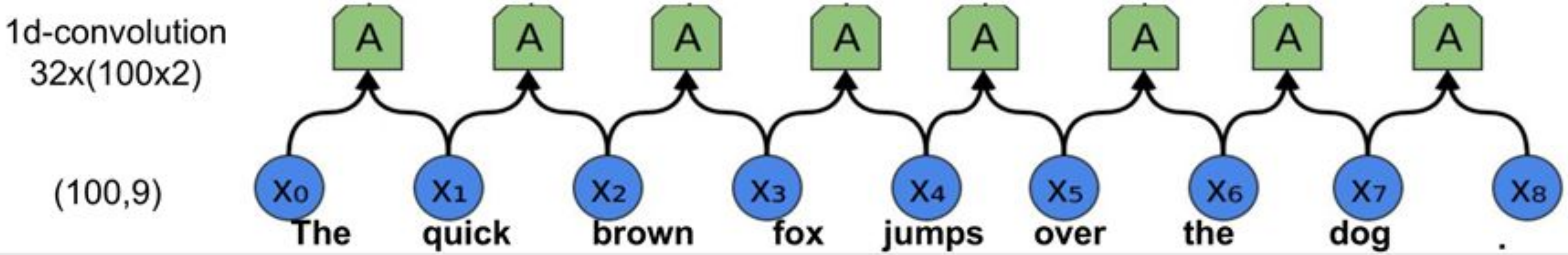
1. Absence of the translational invariance (weights is specific for absolute word coordinate).
2. No parameter sharing.



CNN for texts

CNN for texts

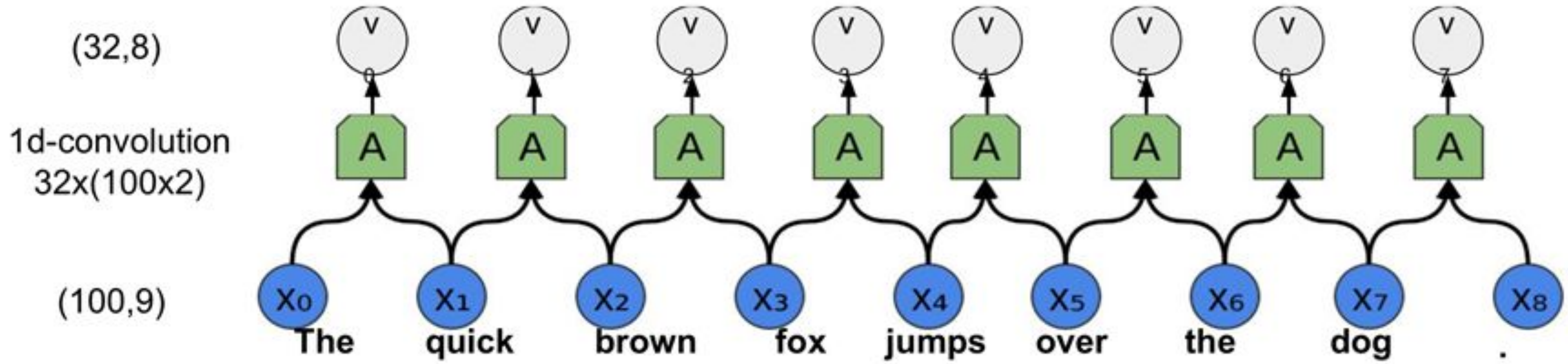
A convolution kernel is a tensor of size
[output dim, embedding dim, kernel size]



CNN for texts

$$\mathbf{v}_0 = \mathbf{A}(\mathbf{x}_0, \mathbf{x}_1)$$

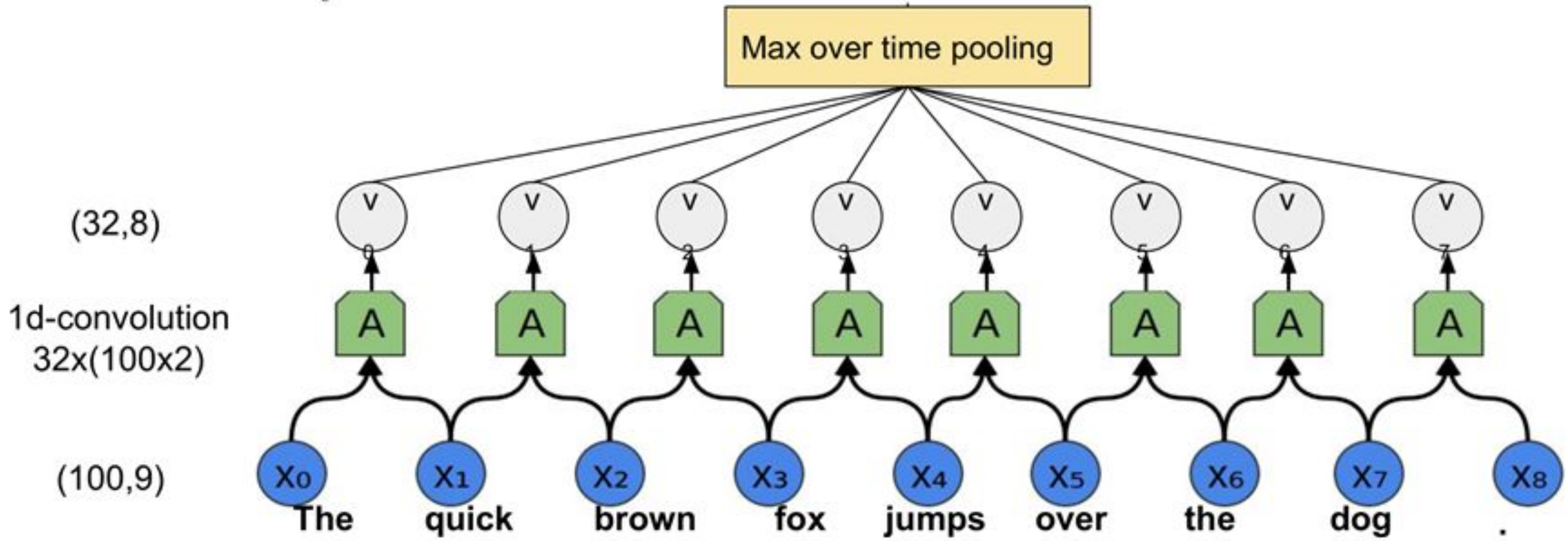
$$v_{0i} = A_i(\mathbf{x}_0, \mathbf{x}_1) = \sum_j (K_{0ij}x_{0j} + K_{1ij}x_{1j})$$



CNN for texts

$$\mathbf{v}_0 = \mathbf{A}(\mathbf{x}_0, \mathbf{x}_1)$$

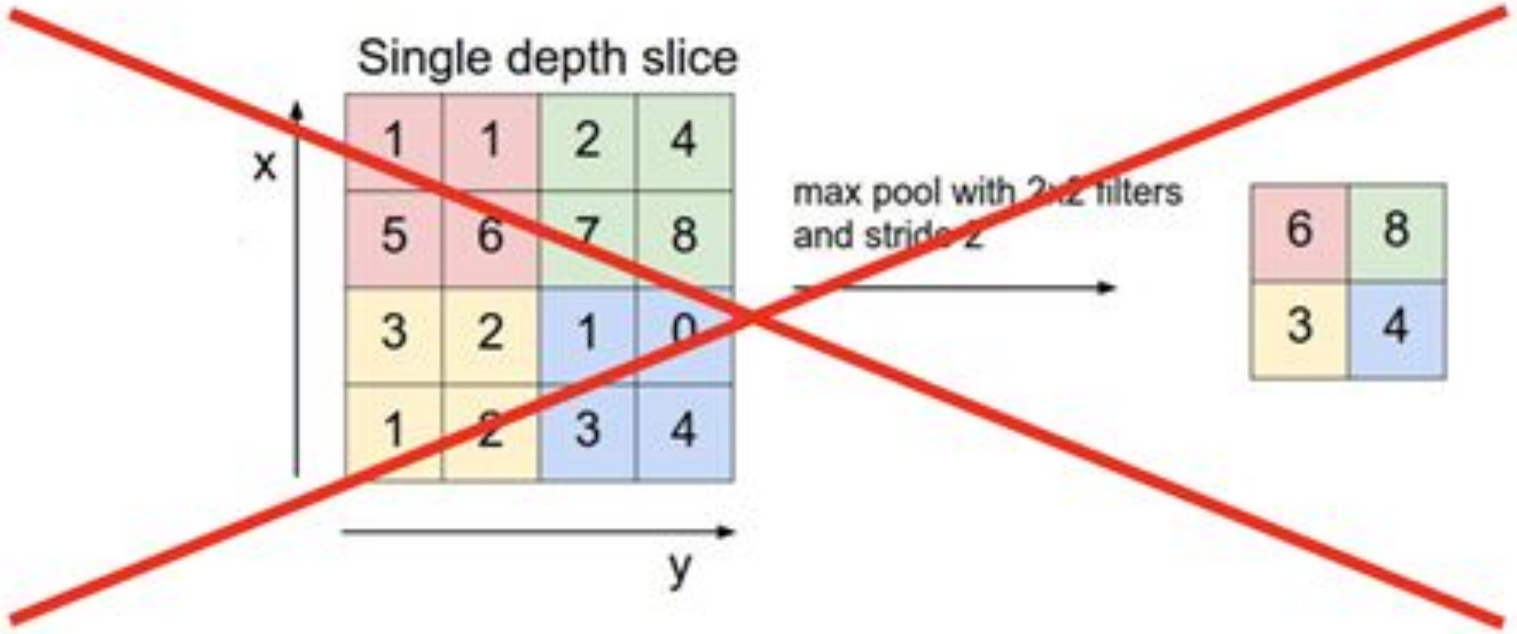
$$v_{0i} = A_i(\mathbf{x}_0, \mathbf{x}_1) = \sum_j (K_{0ij} x_{0j} + K_{1ij} x_{1j})$$

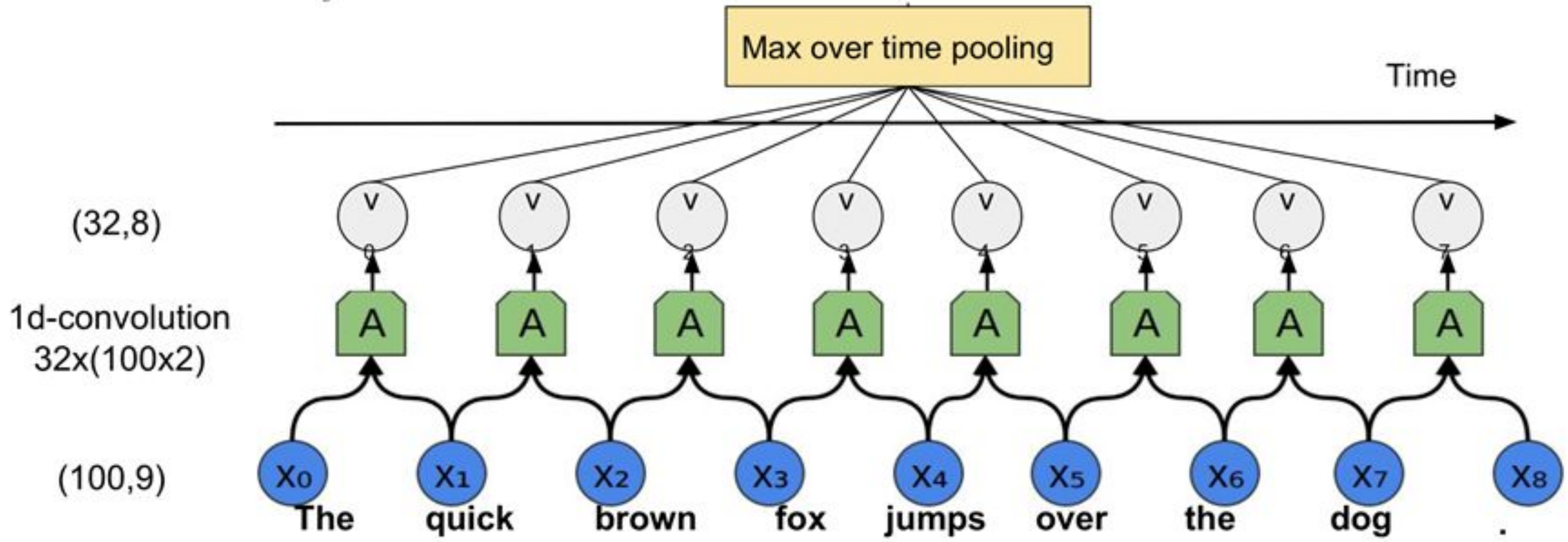


CNN for texts

$$\mathbf{v}_0 = \mathbf{A}(\mathbf{x}_0, \mathbf{x}_1)$$

$$v_{0i} = A_i(\mathbf{x}_0, \mathbf{x}_1) = \sum_j (K_{0ij} x_{0j} + K_{1ij} x_{1j})$$

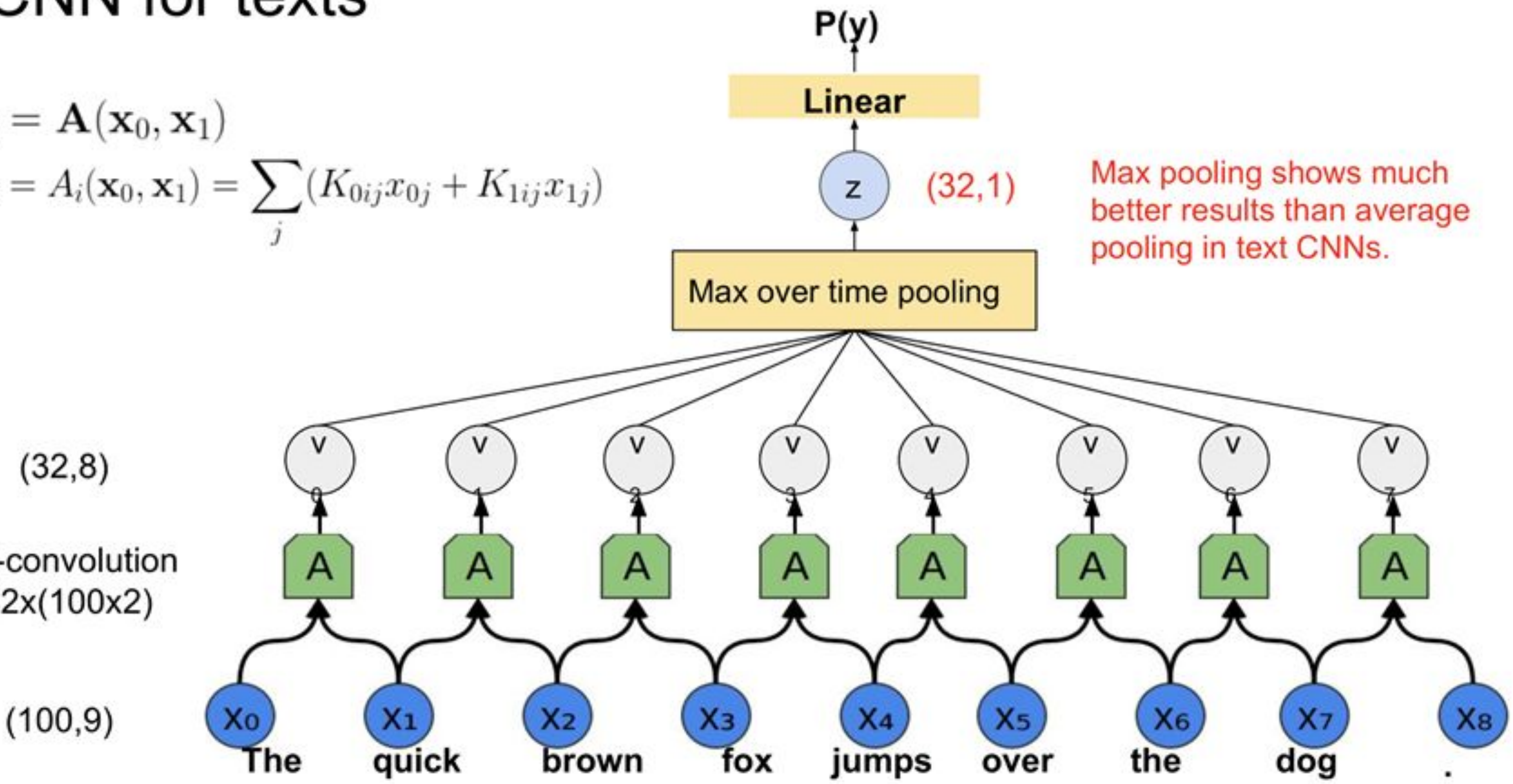




CNN for texts

$$\mathbf{v}_0 = \mathbf{A}(\mathbf{x}_0, \mathbf{x}_1)$$

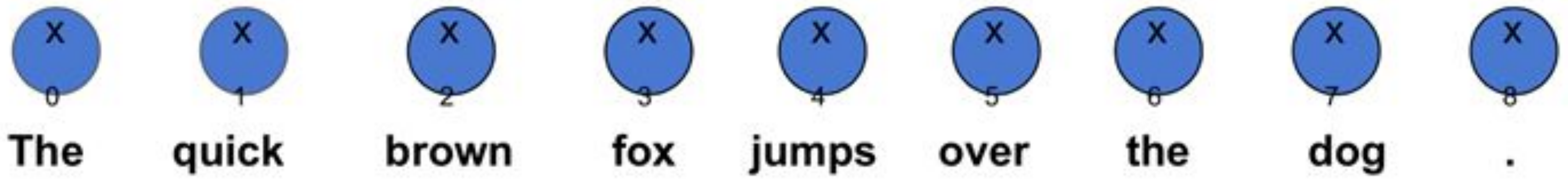
$$v_{0i} = A_i(\mathbf{x}_0, \mathbf{x}_1) = \sum_j (K_{0ij}x_{0j} + K_{1ij}x_{1j})$$



Recurrent NN for text classification

In a RNN Connections between nodes form a directed graph along a sequence.

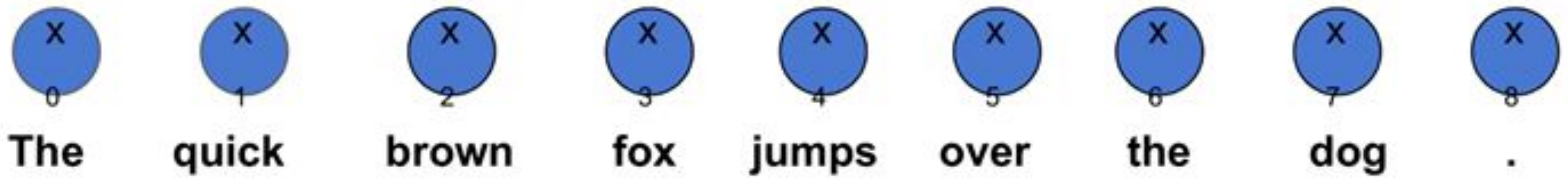
There are different types of recurrent units: vanilla RNN, LSTM, GRU, MI-LSTM, peephole LSTM, …
But it's not important this time.



Recurrent NN for text classification

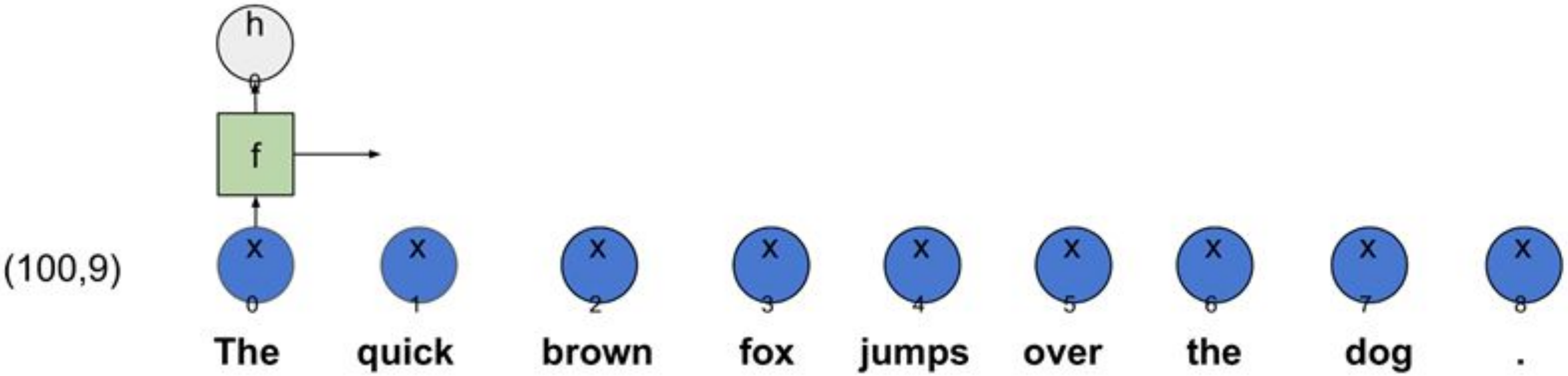
In a RNN Connections between nodes
form a directed graph along a sequence.

$$\mathbf{h}_t = f_w(\mathbf{h}_{t-1}, \mathbf{x}_t)$$



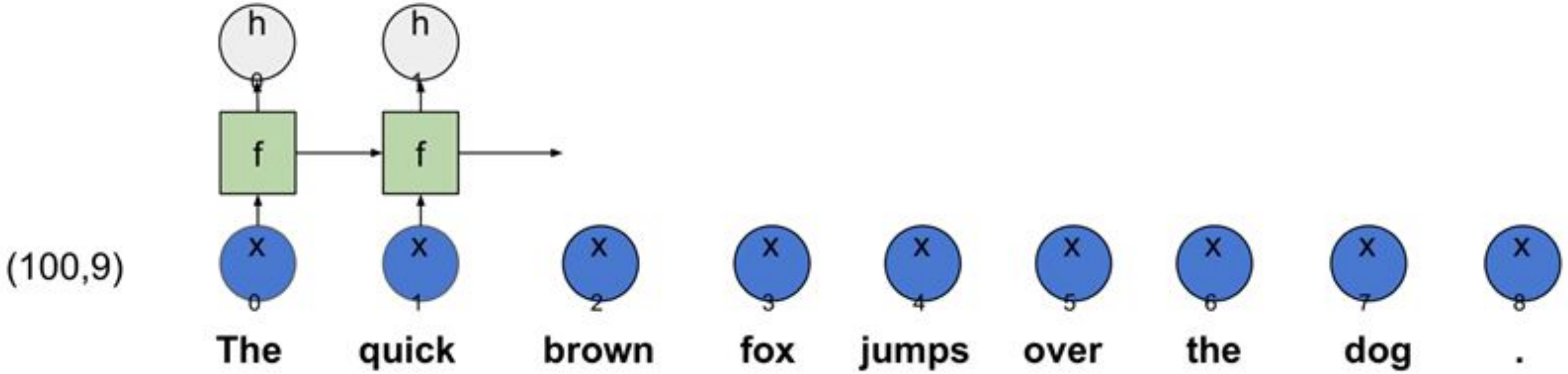
Recurrent NN for text classification

$$\mathbf{h}_t = f_w(\mathbf{h}_{t-1}, \mathbf{x}_t)$$



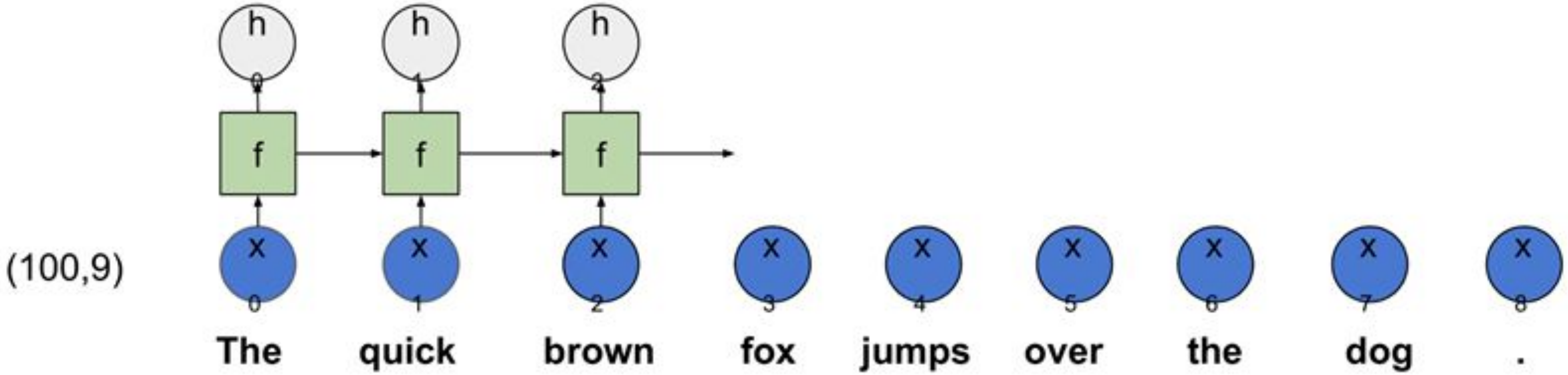
Recurrent NN for text classification

$$\mathbf{h}_t = f_w(\mathbf{h}_{t-1}, \mathbf{x}_t)$$



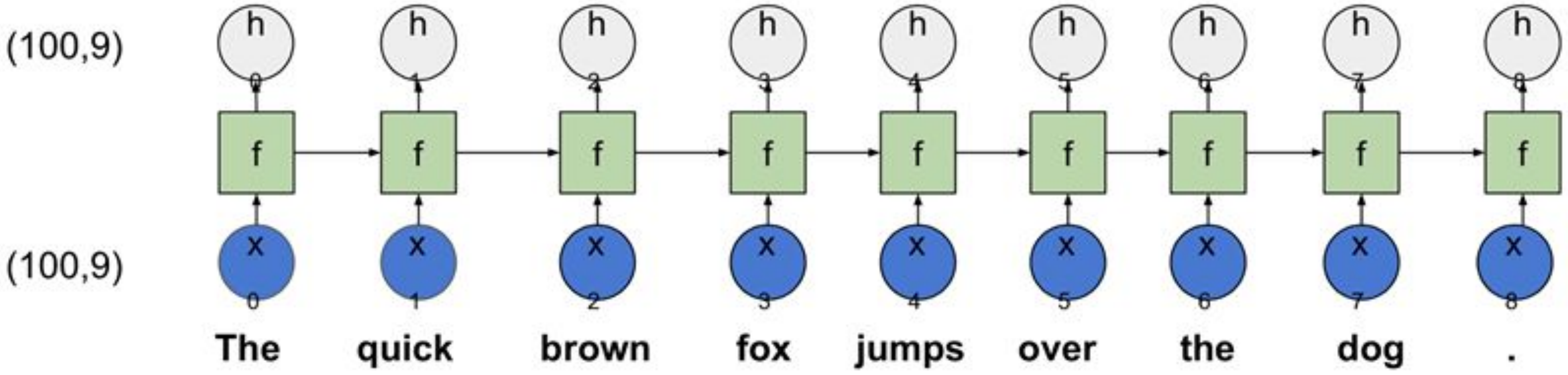
Recurrent NN for text classification

$$\mathbf{h}_t = f_w(\mathbf{h}_{t-1}, \mathbf{x}_t)$$



Recurrent NN for text classification

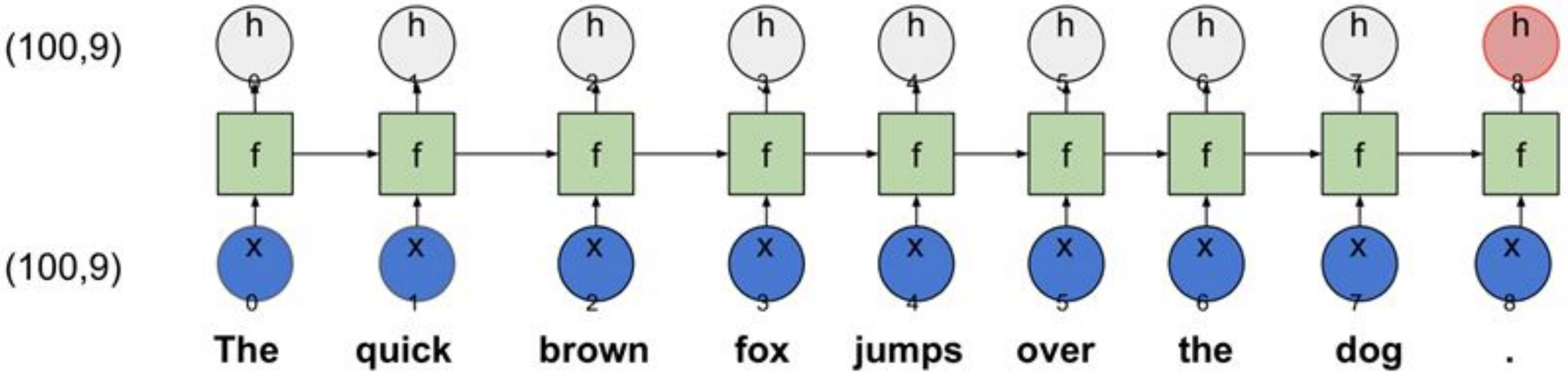
$$\mathbf{h}_t = f_W(\mathbf{h}_{t-1}, \mathbf{x}_t)$$



Recurrent NN for text classification

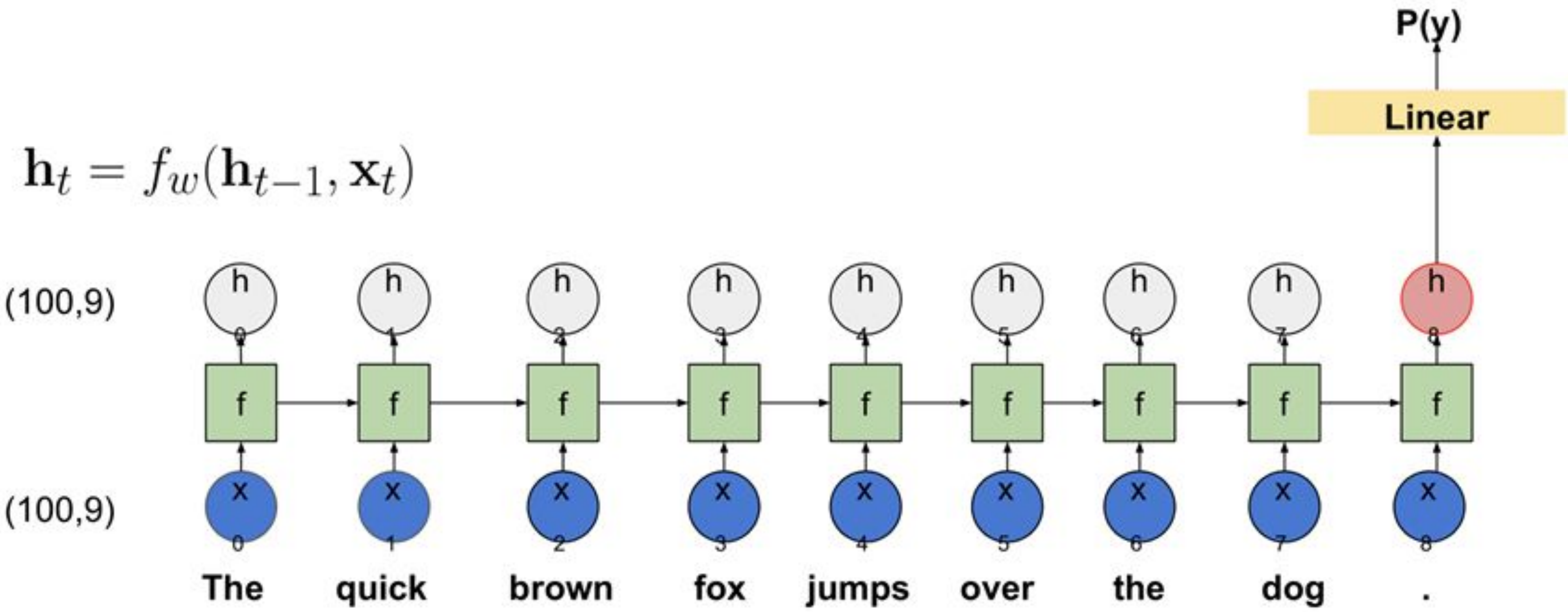
$$\mathbf{h}_8 = f(f(f(\ldots(f(\mathbf{0}, \mathbf{x}_0)), \mathbf{x}_6), \mathbf{x}_7), \mathbf{x}_8)$$

$$\mathbf{h}_t = f_w(\mathbf{h}_{t-1}, \mathbf{x}_t)$$



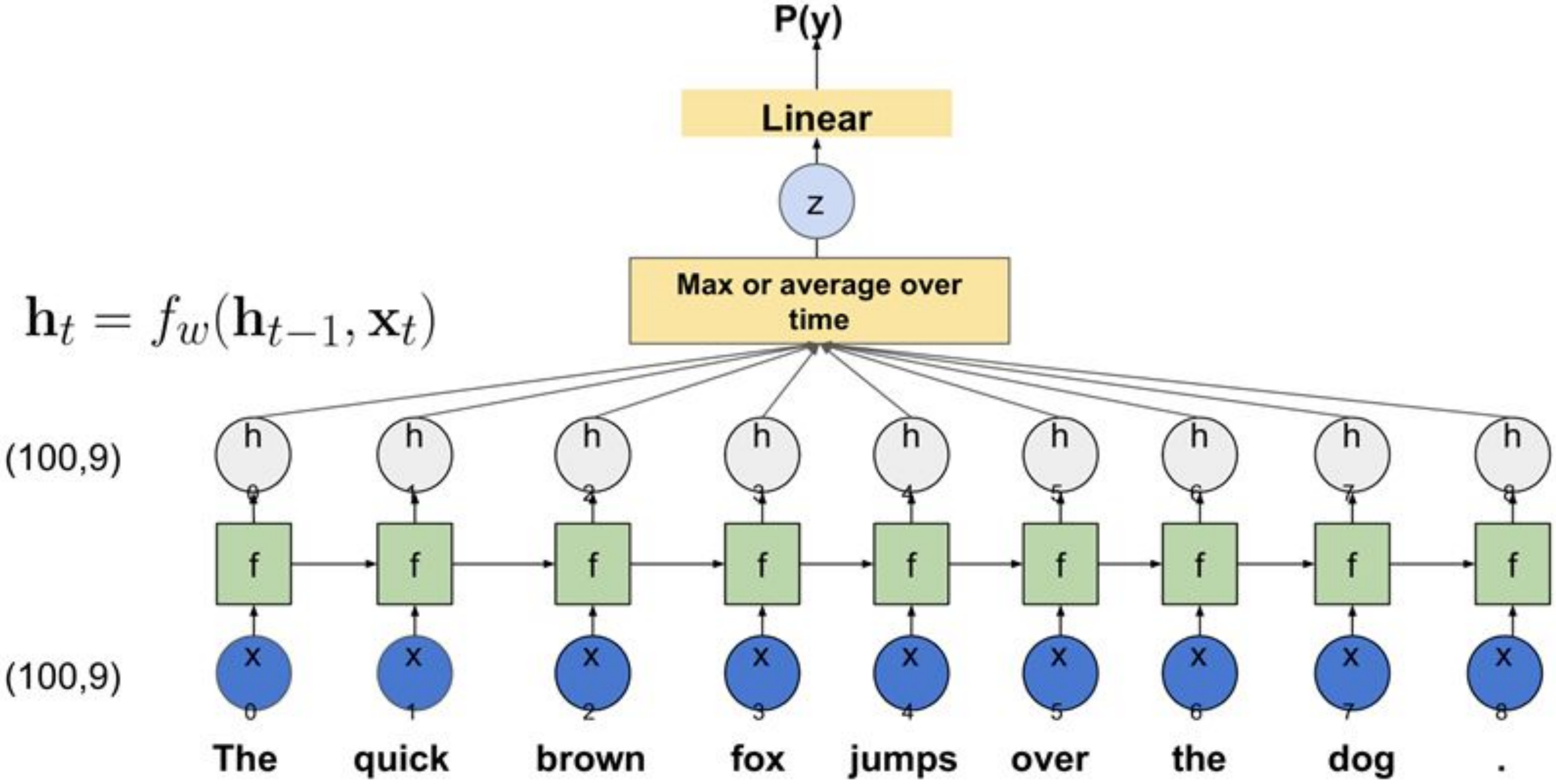
Recurrent NN for text classification

$$\mathbf{h}_t = f_w(\mathbf{h}_{t-1}, \mathbf{x}_t)$$

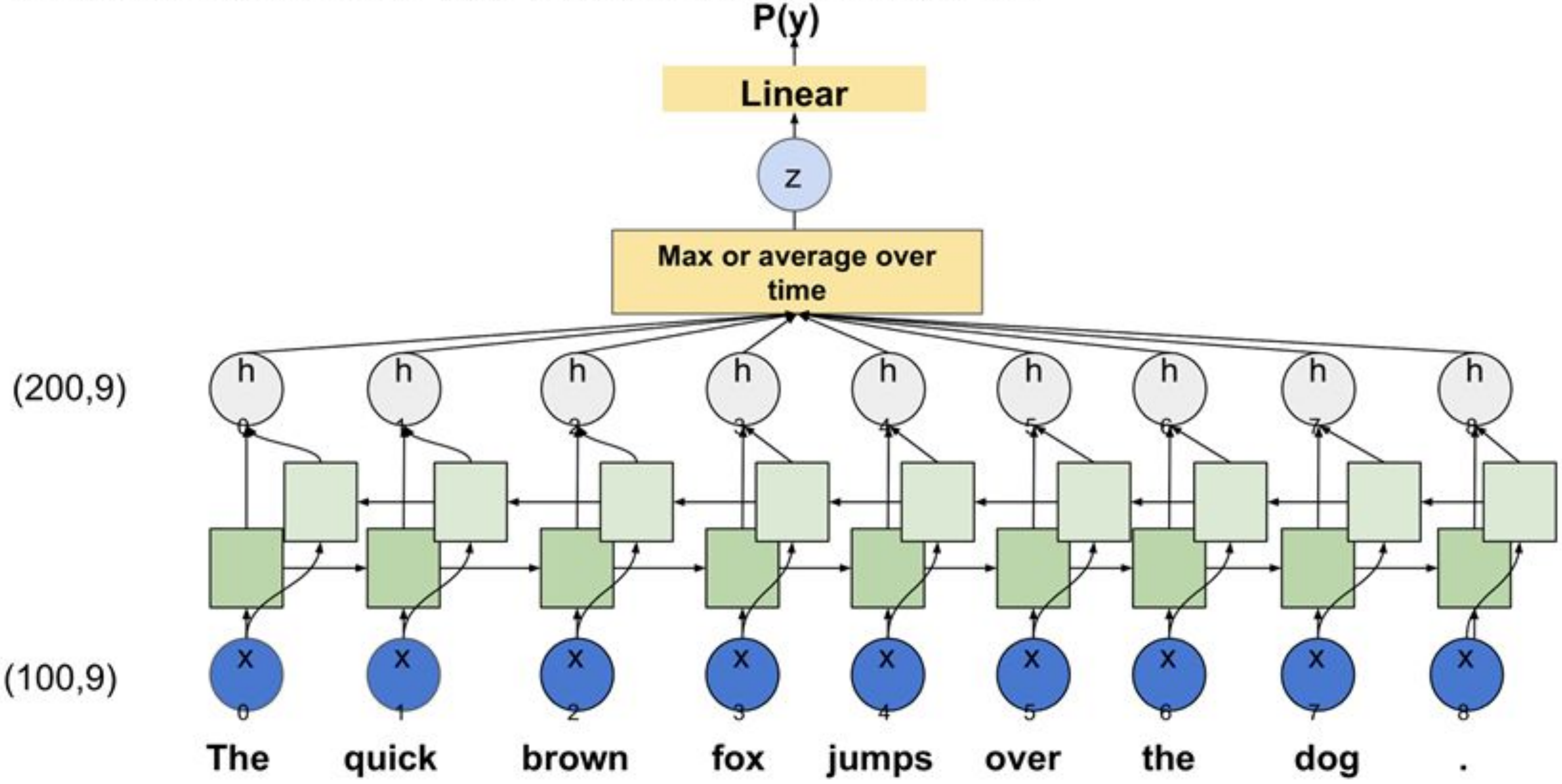


Recurrent NN for text classification

$$\mathbf{h}_t = f_w(\mathbf{h}_{t-1}, \mathbf{x}_t)$$



Recurrent NN for text classification



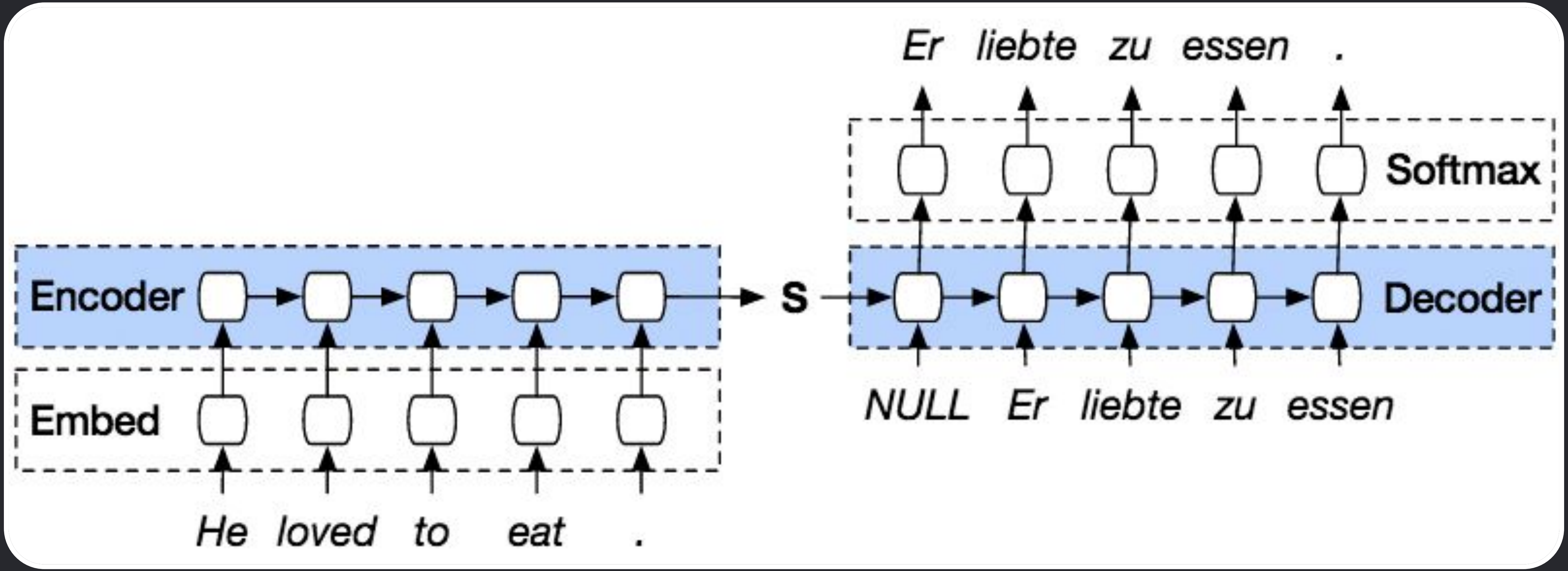
CNNs vs. RNNs

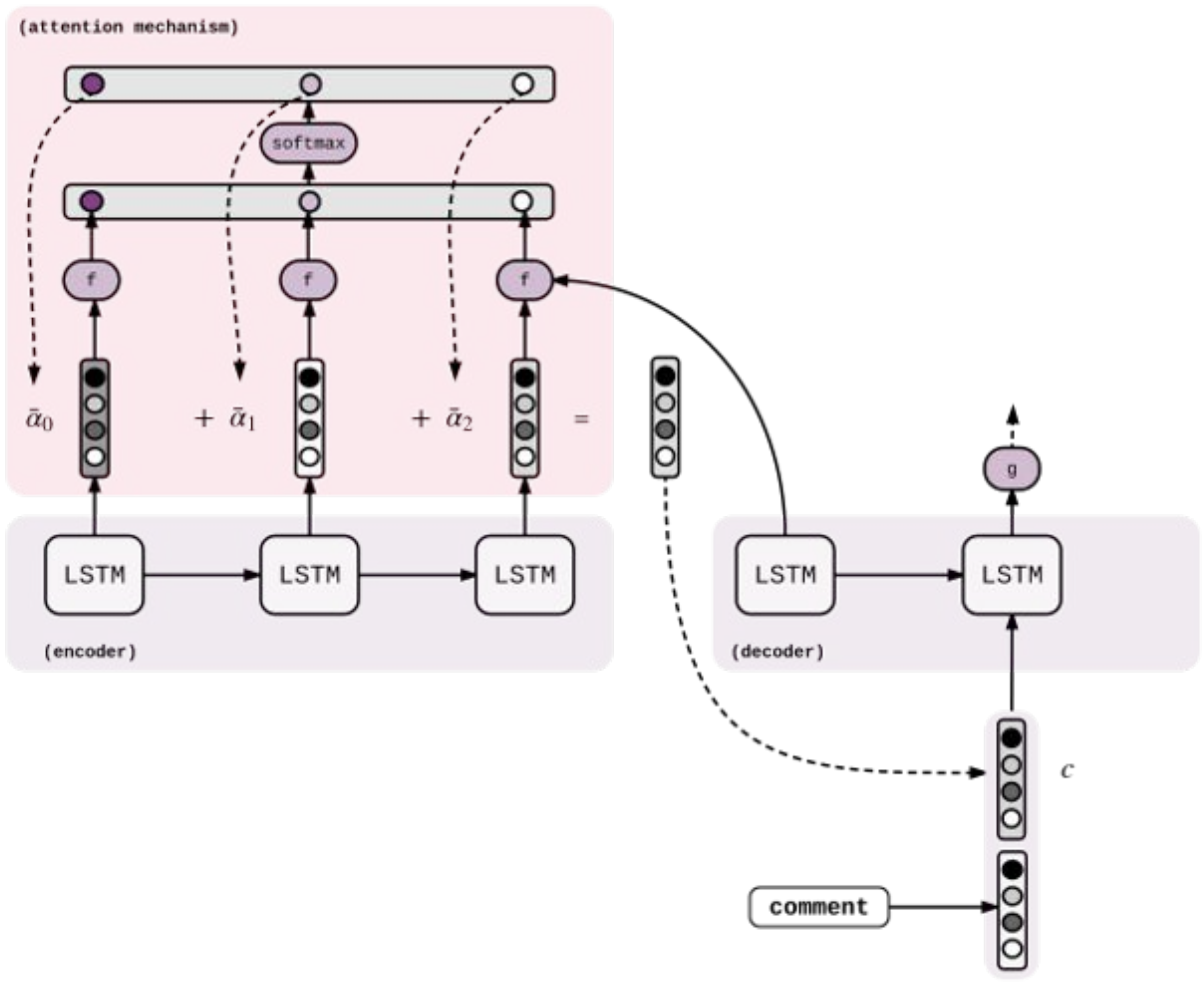
- With a lot of reservations RNNs demonstrates slightly better results on the benchmark classification tasks.
- CNNs work well on the tasks that can be reduced to keyword search. Keyword mean NEs, angry terms and so on.
- Also, RNNs have slower inference than CNNs. CNNs are easier to train.
- For RNN you need more data

It's seems to be very task-dependent thing.
So you should try both options.

[Yin et al 2017]

Beyond text classification


Er liebte zu essen .
Softmax
Encoder
S
Decoder
Embed
NULL Er liebte zu essen
He loved to eat .


(attention mechanism)
softmax
f
f
f
$\bar{\alpha}_0$
$+\ \bar{\alpha}_1$
$+\ \bar{\alpha}_2$
=
LSTM
LSTM
LSTM
(encoder)
LSTM
LSTM
(decoder)
g
c
comment

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.

Спасибо за внимание!