# Natural Language Processing: Classification

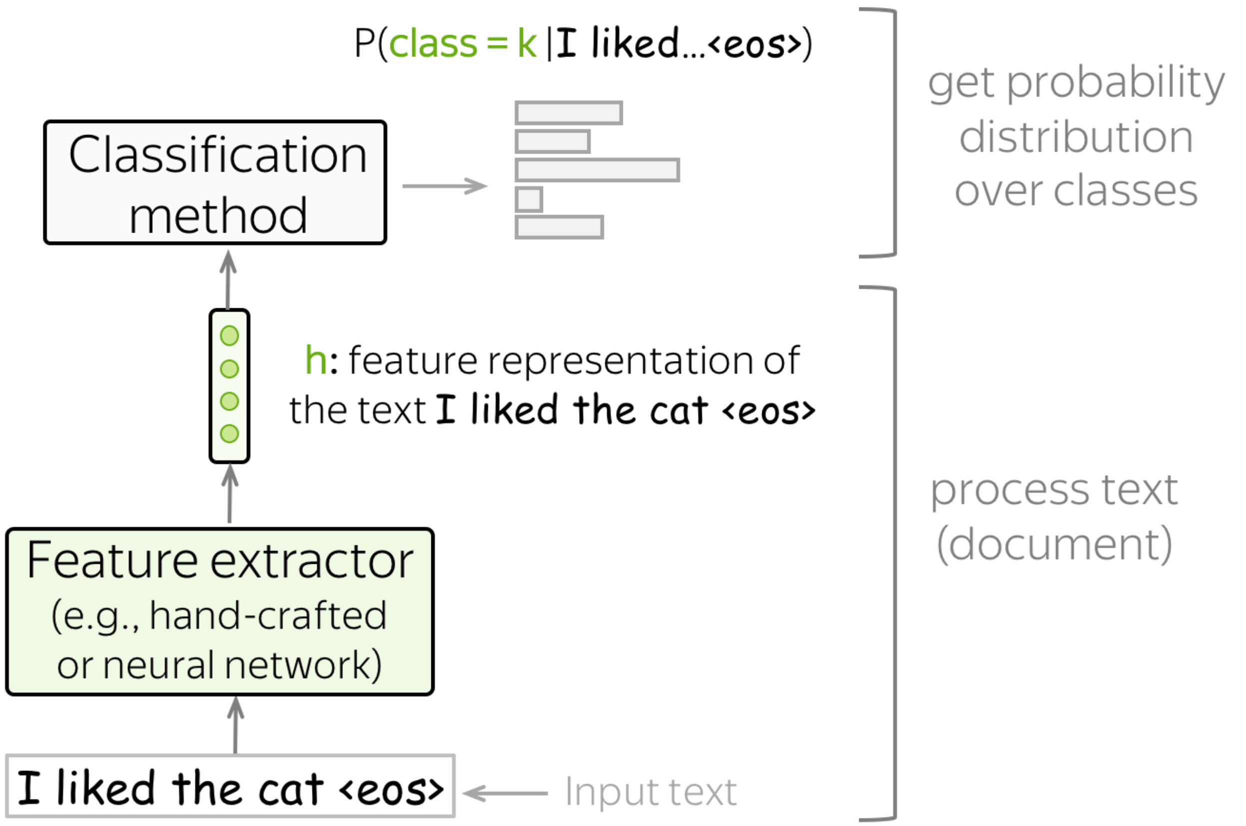
HSE Faculty of Computer Science

Machine Learning and Data-Intensive Systems

Murat Khazhgeriev

# Table of Content

# Extract features, apply model, compare the distribution to the target




Source: https://lena-voita.github.io/nlp_course/text_classification.html

# Table of Content

# Naively count the conditional probability

Bayes' rule
(hence Naïve Bayes)

Ignore P(x) – it does not
influence the argmax

$$y^* = \arg\max_k P(y=k|x) = \arg\max_k \frac{P(x|y=k)\cdot P(y=k)}{P(x)} = \arg\max_k \underbrace{P(x|y=k)}_{} \cdot \underbrace{P(y=k)}_{}$$

need to define this

Source: https://lena-voita.github.io/nlp_course/text_classification.html

# Naively count the conditional probability

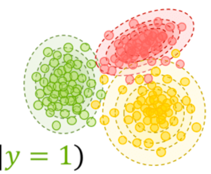
The Naive Bayes assumptions are

•Bag of Words assumption: word order does not matter

•Conditional Independence assumption: features (words) are independent given the class

$$y^* = \arg\max_k \underbrace{P(y = k|x)} = \arg\max_k \underbrace{P(x|y = k)} \cdot \underbrace{P(y = k)} = \arg\max_k \underbrace{P(x, y = k)}$$

posterior probability: after looking at data (i.e., we know **x**)



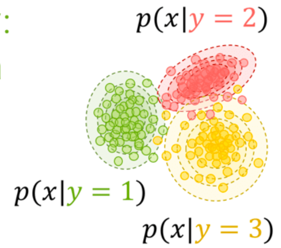

prior probability: before looking at data (i.e., we don't know **x**)

joint probability (hence the model is **generative**)

Source: https://lena-voita.github.io/nlp_course/text_classification.html

# How to define P(x|y=k) and P(y=k)

$$y^* = \arg\max_k \underbrace{P(y=k|x)}_{\substack{\text{posterior probability:}\\ \text{after looking at data}\\ \text{(i.e., we know } \mathbf{x}\text{)}}} = \arg\max_k \underbrace{P(x|y=k)} \cdot \underbrace{P(y=k)}_{\substack{\text{prior probability:}\\ \text{before looking at data}\\ \text{(i.e., we don't know } \mathbf{x}\text{)}}} = \arg\max_k \underbrace{P(x, y=k)}_{\substack{\text{joint probability}\\ \text{(hence the model}\\ \text{is } \textbf{generative}\text{)}}}$$




$$P(y=k) = \frac{N(y=k)}{\sum_i N(y=i)}$$

$$P(x|y=k) = P(x_1, \dots, x_n|y=k) = \prod_{t=1}^{n} P(x_t|y=k)$$

$$P(x_i|y=k) = \frac{N(x_i, y=k)}{\sum_{t=1}^{|V|} N(x_t, y=k)}$$

Source: https://lena-voita.github.io/nlp_course/text_classification.html

# How to define P(x | y=k) and P(y=k)?

The Naive Bayes assumptions are

•Bag of Words assumption: word order does not matter,

•Conditional Independence assumption: features (words) are independent given the class.

$$P(y=k) = \frac{N(y=k)}{\sum_i N(y=i)}$$

$$P(x|y=k) = P(x_1, \dots, x_n|y=k) = \prod_{t=1}^{n} P(x_t|y=k)$$

Source: https://lena-voita.github.io/nlp_course/text_classification.html

## What if $N(x_i, y = k) = 0$? Need to avoid this!

nulls out token prob. $\Longrightarrow$ nulls out document probability $\Longrightarrow$ Bad!

$$N(x_i, y = k) \Longrightarrow P(x_i|y = k) = \frac{N(x_i, y = k)}{\sum_{t=1}^{|V|} N(x_t, y = k)} = 0 \Longrightarrow P(x|y = k) = \prod_{i=1}^{n} P(x_i|y = k) = 0$$

In training data, haven't seen token $x_i$ in documents of class $k$

$$P(x_i|y = k) = \frac{\delta + N(x_i, y = k)}{\sum_{t=1}^{|V|} (\delta + N(x_t, y = k))} = \frac{\delta + N(x_i, y = k)}{\delta \cdot |V| + \sum_{t=1}^{|V|} N(x_t, y = k)}$$

Source: https://lena-voita.github.io/nlp_course/text_classification.html

# Making a prediction

Data: $x =$ This film is awesome !
($x_1$ = This, $x_2$ = film, $x_3$ = is, $x_4$ = awesome, $x_5$ = !)

Compute joint probability of data and class

**Positive class**

$$P(x, y=+) = P(y=+)\cdot P(x|y=+) = P(y=+)\cdot P(\text{This}|y=+)\cdot P(\text{film}|y=+)\cdot P(\text{is}|y=+)\cdot \boxed{P(\text{awesome}\,|y=+)}\cdot P(!|y=+)$$

**Negative class**

$$P(x, y=-) = P(y=-)\cdot P(x|y=-) = P(y=-)\cdot P(\text{This}|y=-)\cdot P(\text{film}|y=-)\cdot P(\text{is}\,|y=-)\cdot \boxed{P(\text{awesome}\,|y=-)}\cdot P(!\,|y=-)$$

Prior class probability (often 0.5)

Neutral words – not much difference in probabilities for classes

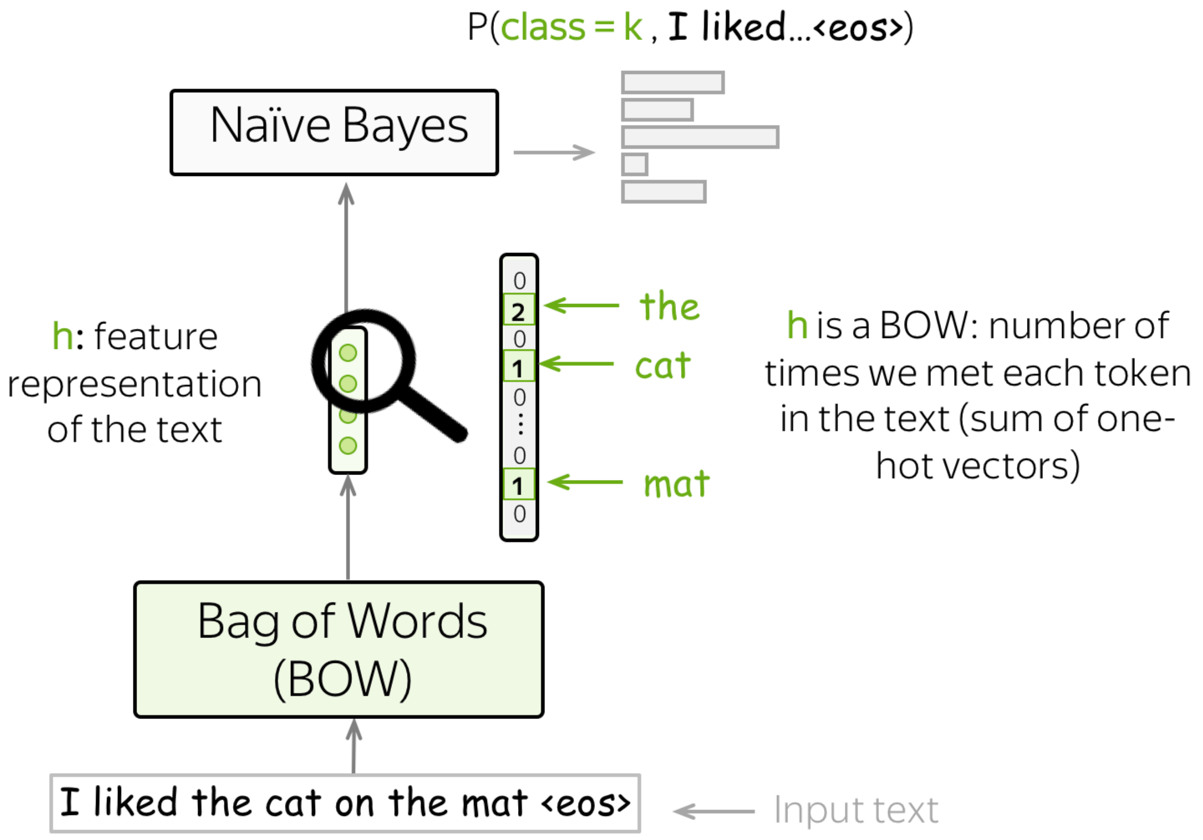
This is where we expect the difference:

$$P(\text{awesome}\,|y=+) \gg P(\text{awesome}|y=-)$$

$$P(x, y=+) > P(x, y=-) \Rightarrow y = +$$

Source: https://lena-voita.github.io/nlp_course/text_classification.html

# View in the general framework

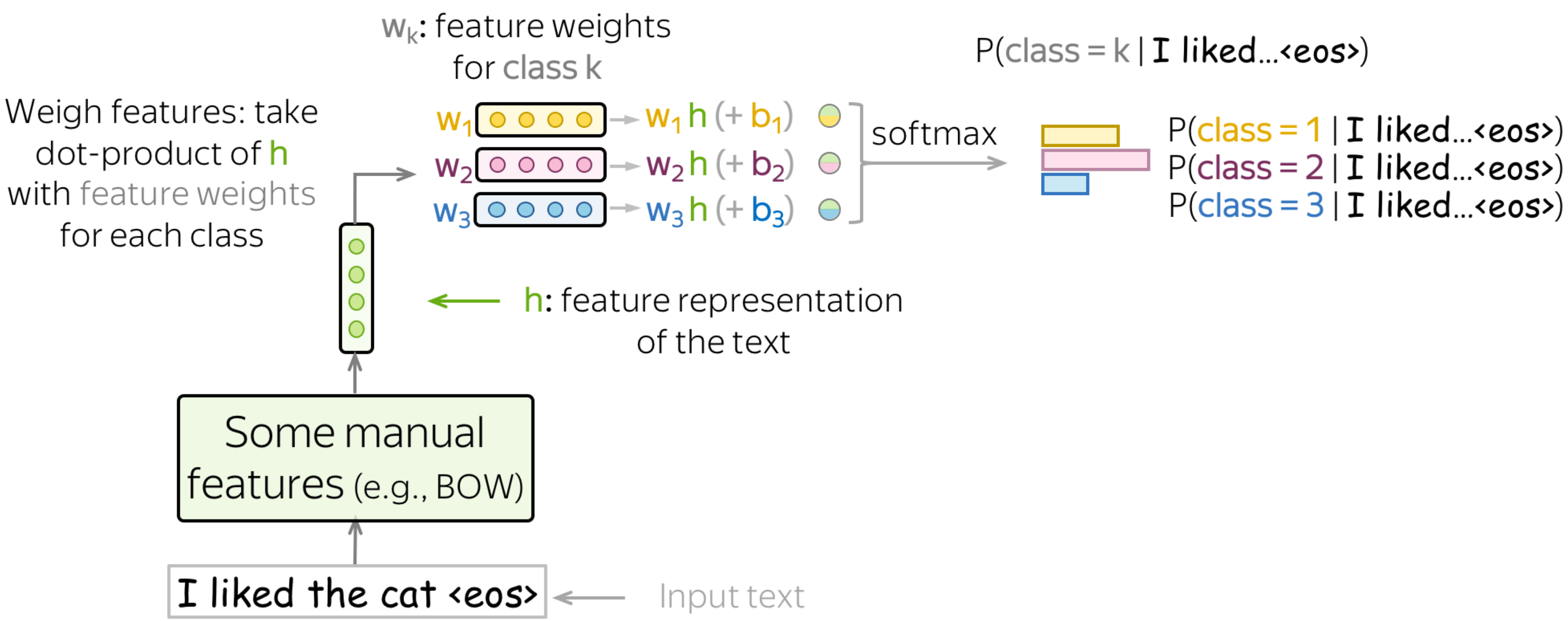



Source: https://lena-voita.github.io/nlp_course/text_classification.html

# Table of Content

- General view on classification with text
- **Pre-deep learning approaches: Logistic Regression**
- Deep learning approaches
- Practical Tips

# Train a logistic regression




Source: https://lena-voita.github.io/nlp_course/text_classification.html

# Loss function: log-loss

Given training examples $x^1, \dots, x^N$ with labels $y^1, \dots, y^N$, $y^i \in \{1, \dots, K\}$, we pick those weights $w^{(k)}, k = 1..K$ which maximize the probability of the training data:

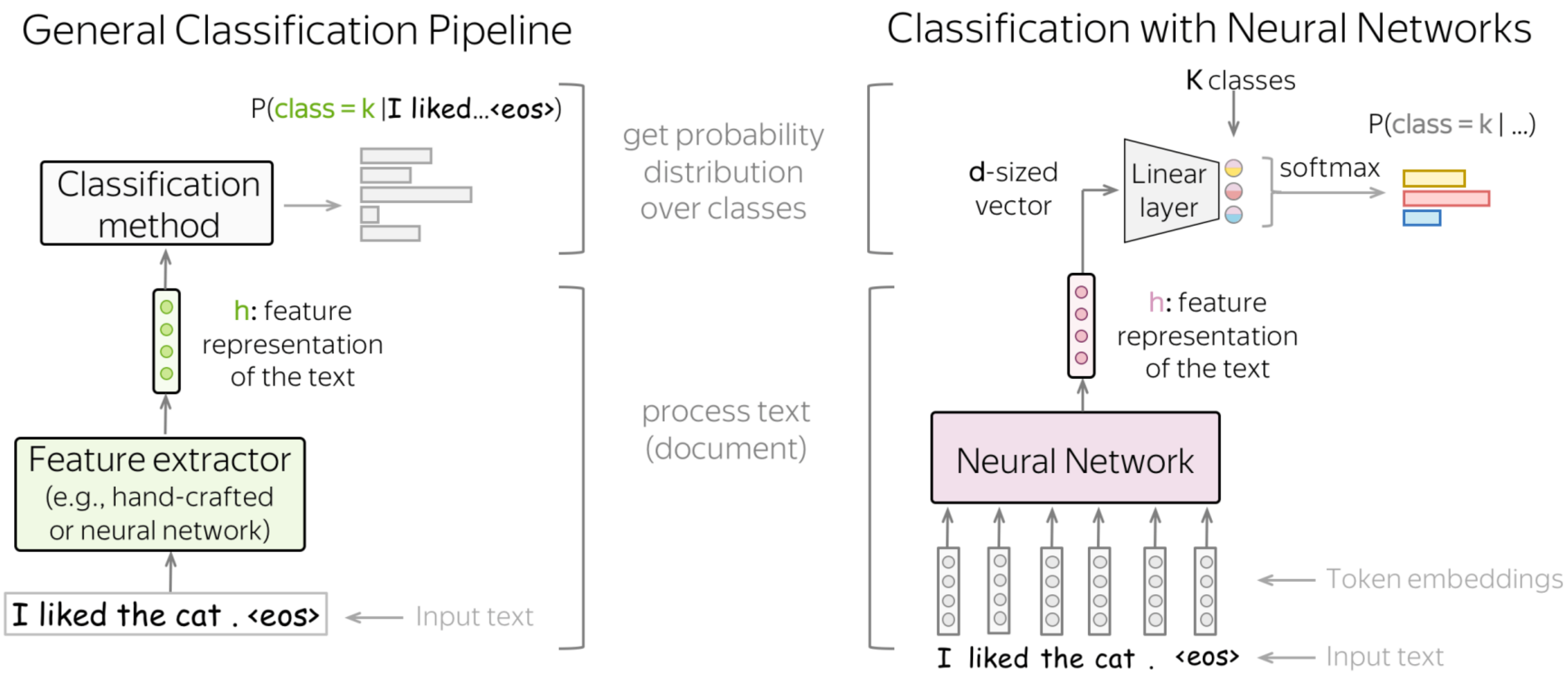
$$w^* = \arg\max_w \sum_{i=1}^{N} \log P(y = y^i | x^i).$$

Source: https://lena-voita.github.io/nlp_course/text_classification.html

# Table of Content

- General view on classification with text
- Pre-deep learning approaches
- **Deep learning approaches: Overview**
- Practical Tips

# Overview of DL-based approaches




Source: https://lena-voita.github.io/nlp_course/text_classification.html

# Overview of DL-based approaches

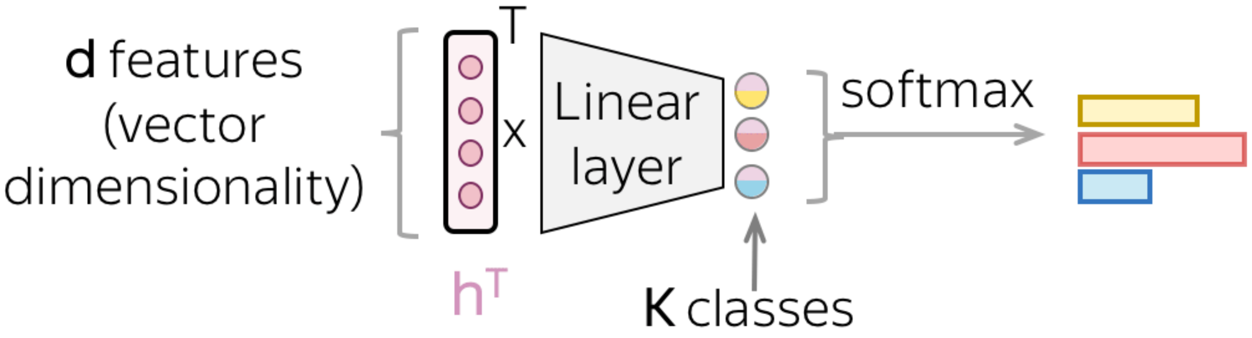
We have:

- $h$ - vector of size **d**

We need:

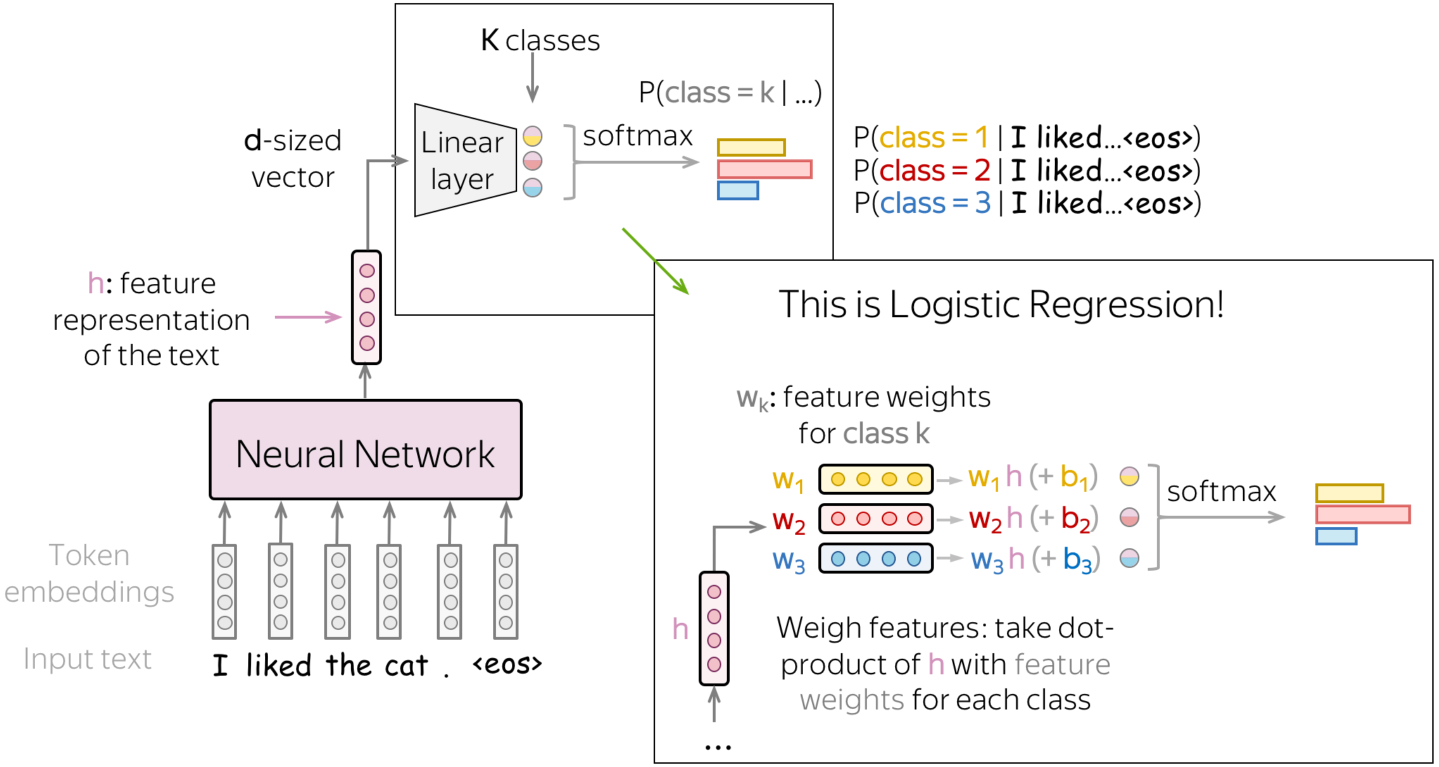
- vector of size **K** – probabilities for K classes

→ Transform linearly from size **d** to size **K**



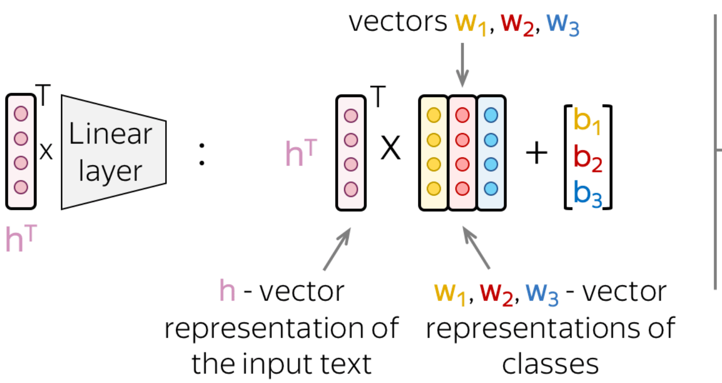

Source: https://lena-voita.github.io/nlp_course/text_classification.html

# Overview of DL-based approaches



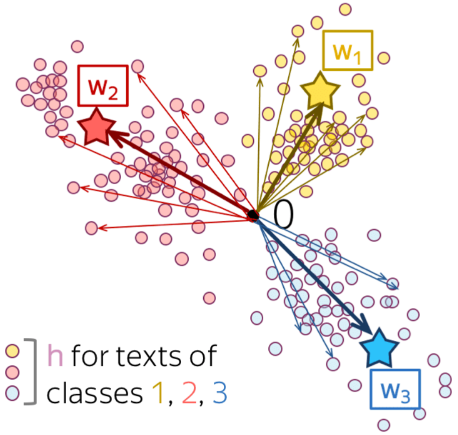

Source: https://lena-voita.github.io/nlp_course/text_classification.html

# Overview of DL-based approaches



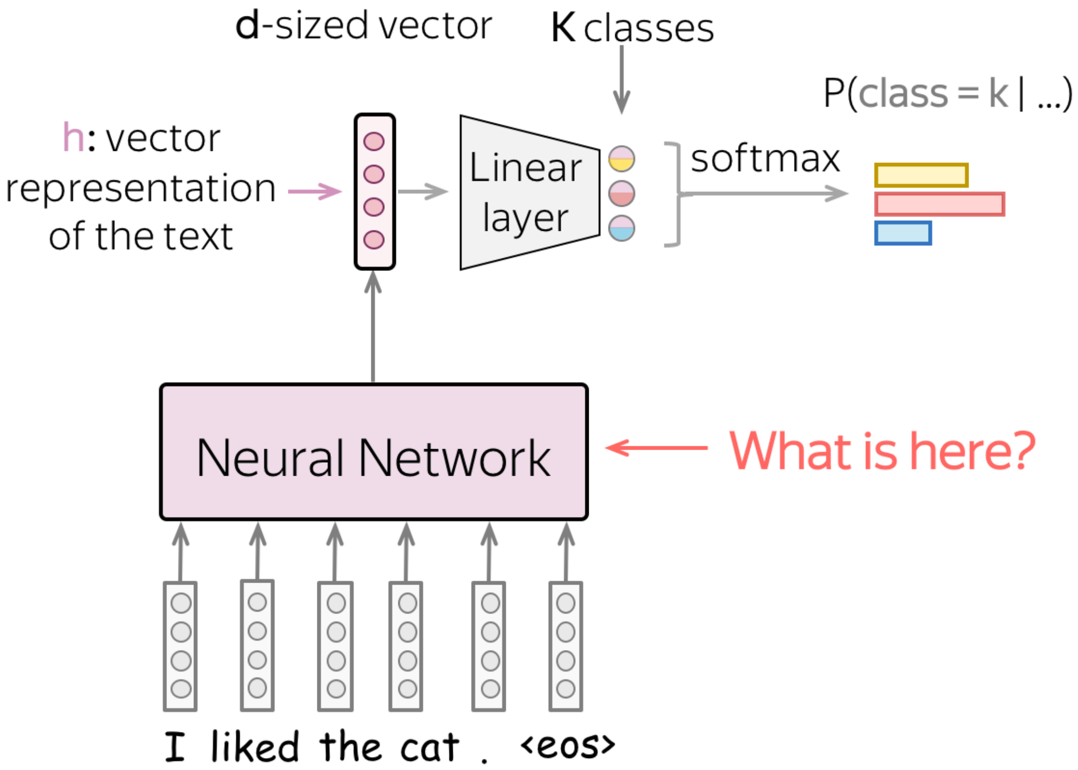

What NN learns (hopefully):

Text vectors point in the direction of the corresponding class vectors




Source: https://lena-voita.github.io/nlp_course/text_classification.html

# Overview of DL-based approaches

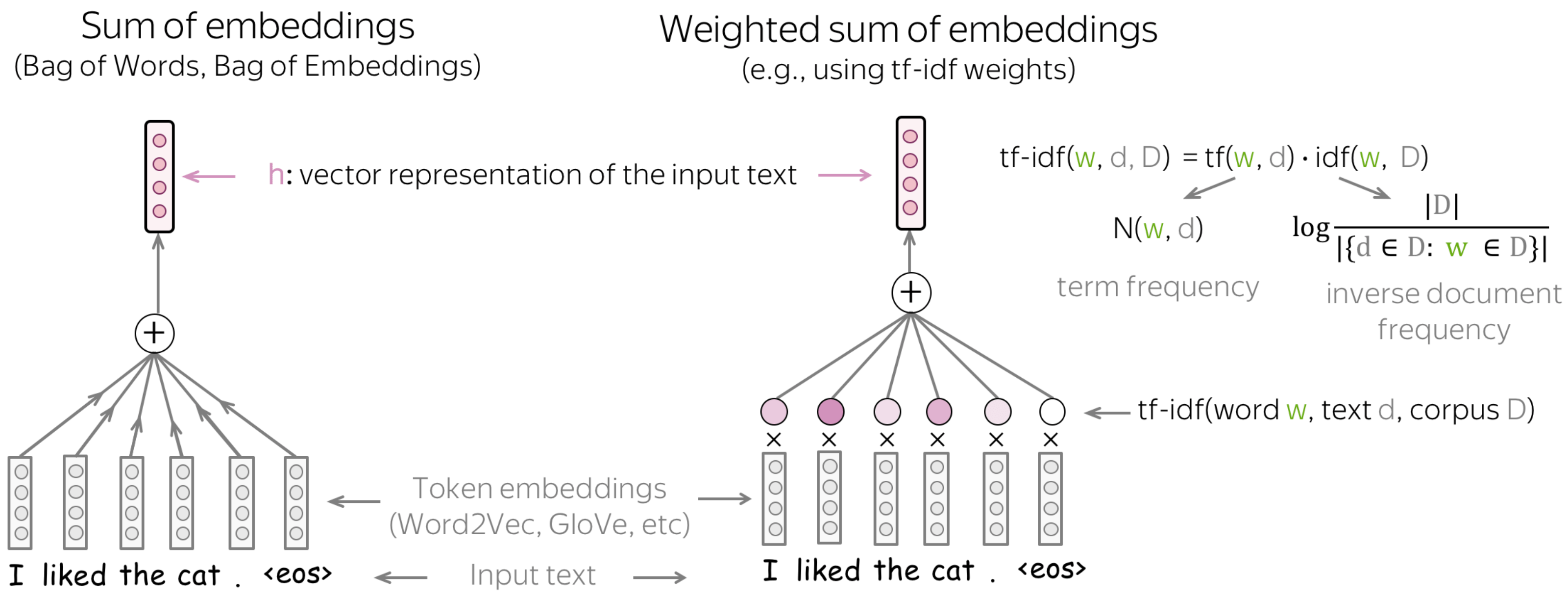



Source: https://lena-voita.github.io/nlp_course/text_classification.html

# Table of Content

- General view on classification with text
- Pre-deep learning approaches
- **Deep learning approaches: Basics**
- Practical Tips

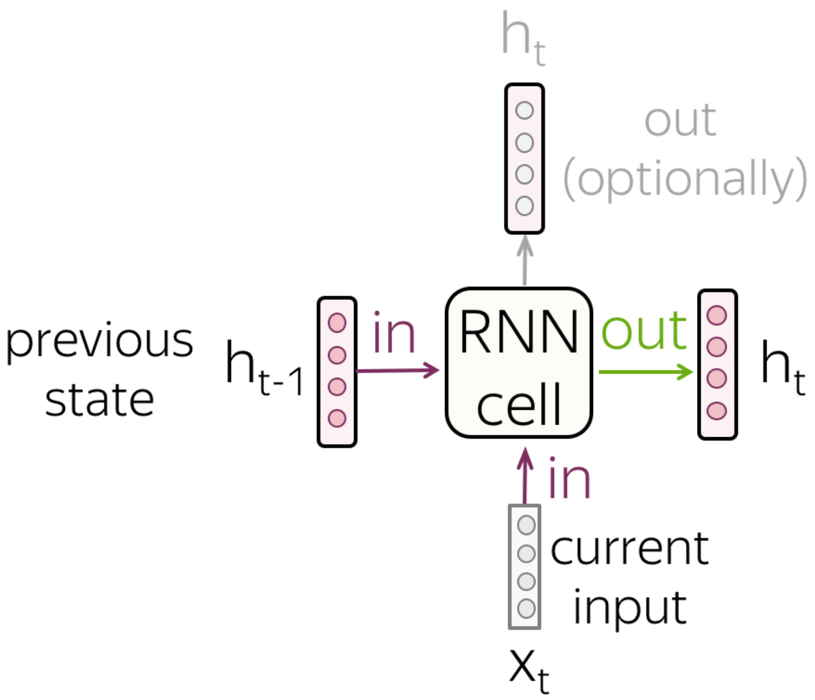
# Vanilla embeddings




Source: https://lena-voita.github.io/nlp_course/text_classification.html

# Table of Content

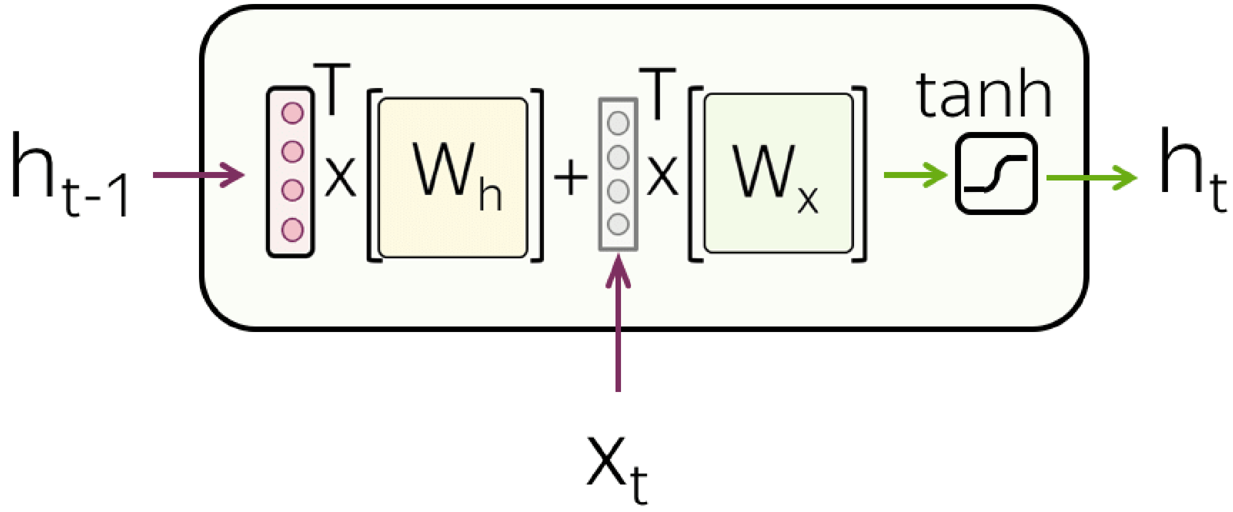
# Vanilla embeddings




Source: https://lena-voita.github.io/nlp_course/text_classification.html

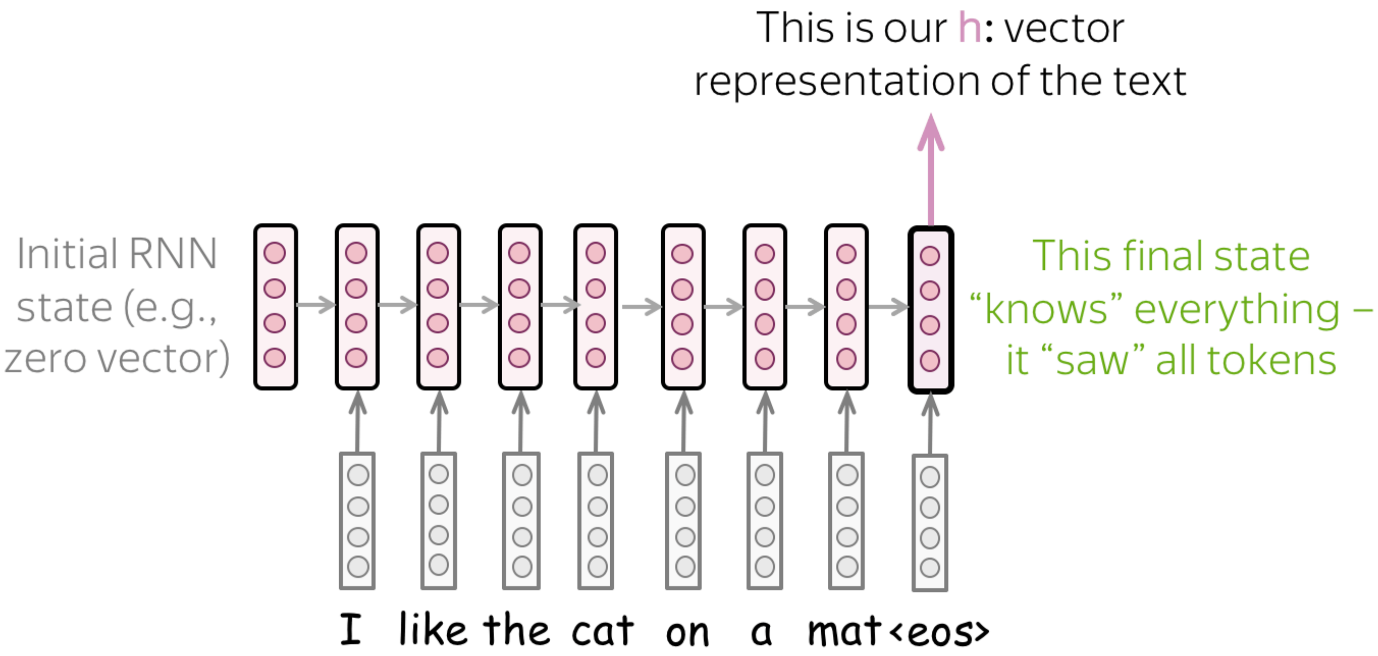
# Vanilla RNN

$$h_t = \tanh(h_{t-1}W_h + x_t W_x)$$



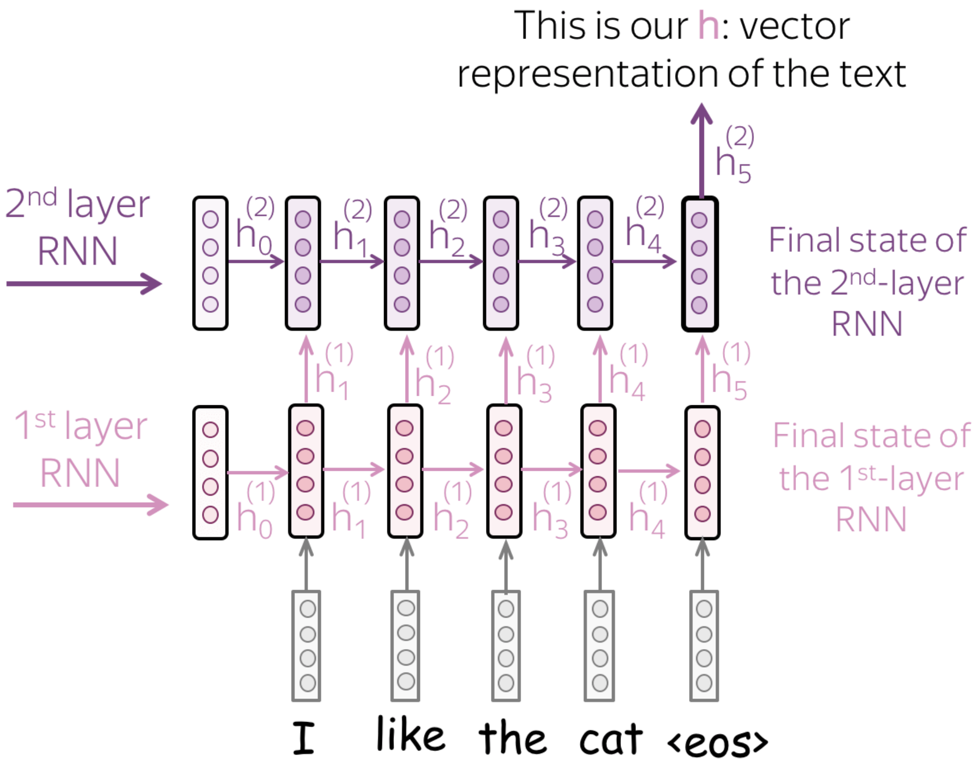

Source: https://lena-voita.github.io/nlp_course/text_classification.html

# Vanilla RNN



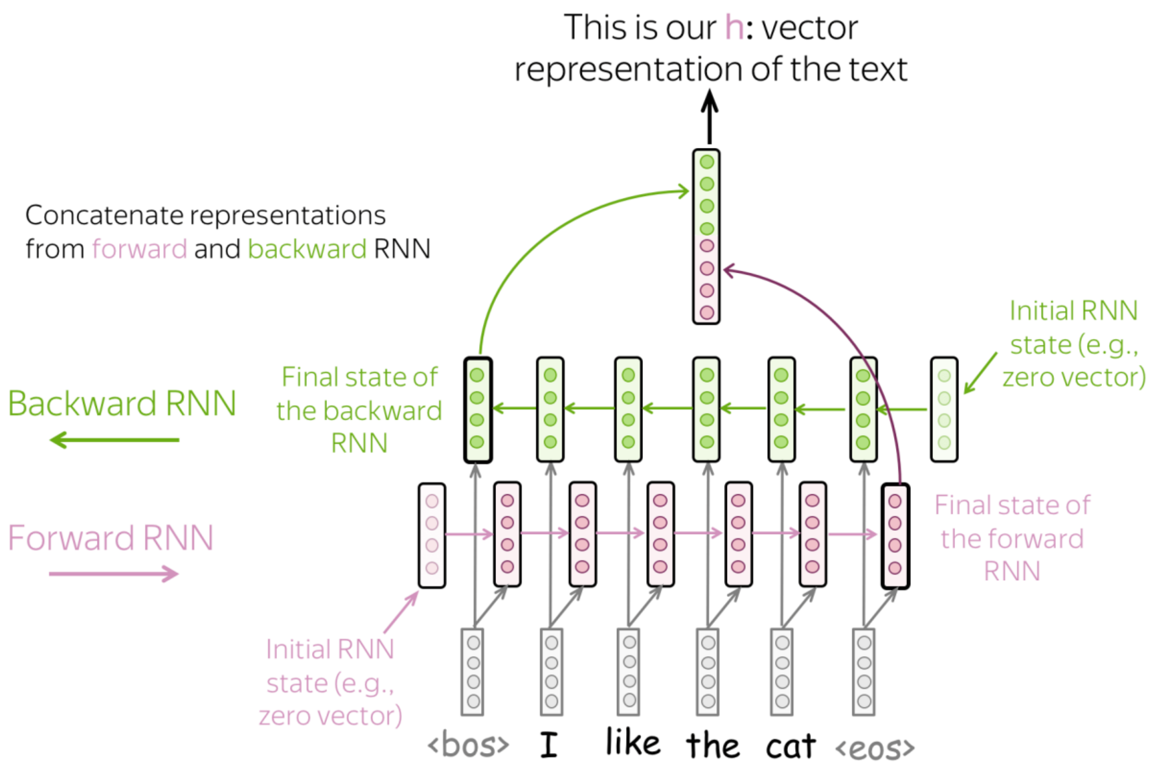

Source: https://lena-voita.github.io/nlp_course/text_classification.html

# Stacked RNN




Source: https://lena-voita.github.io/nlp_course/text_classification.html

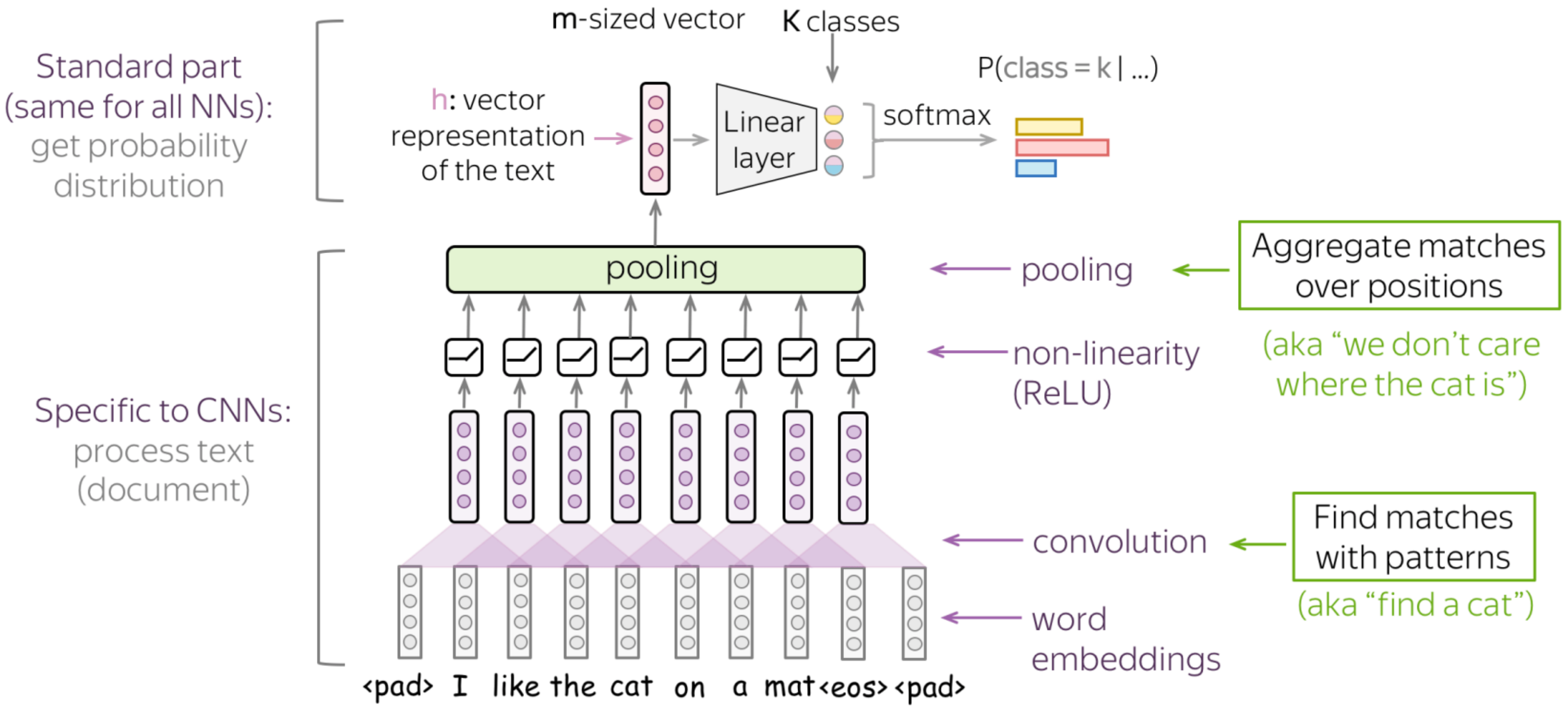
# Bidirectional RNN




Source: https://lena-voita.github.io/nlp_course/text_classification.html

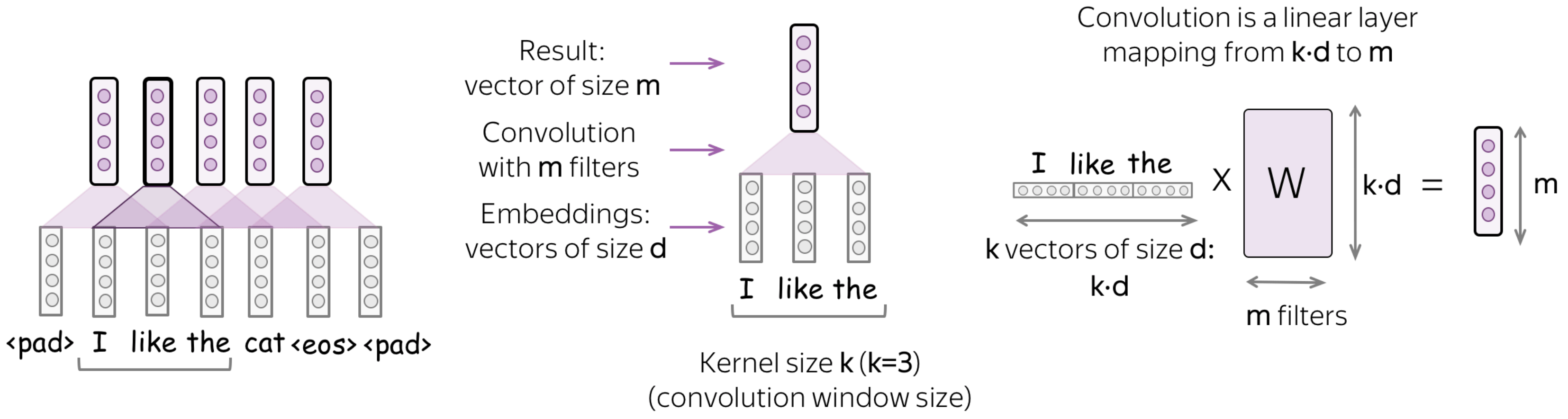
# Table of Content

# CNN overview



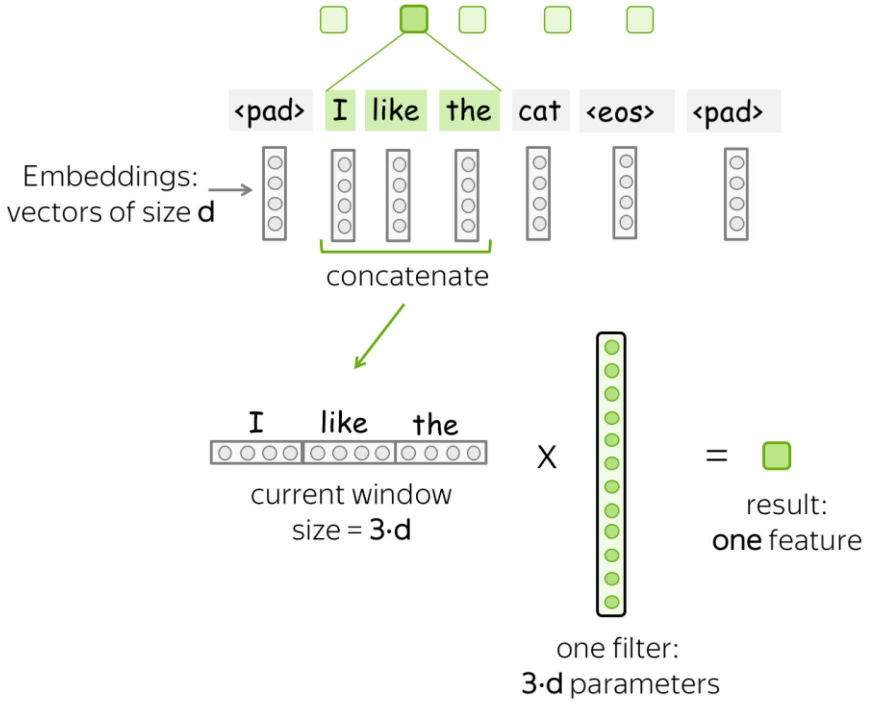

Source: https://lena-voita.github.io/nlp_course/text_classification.html

# Convolution is a Linear Operation Applied to Each Window



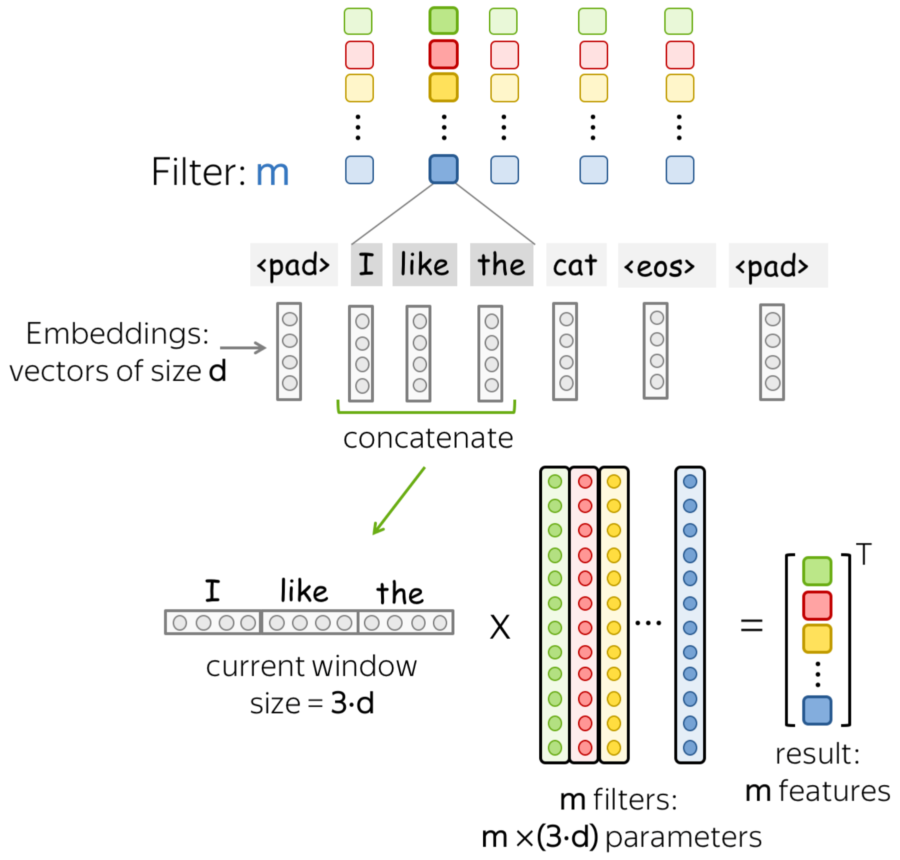

Source: https://lena-voita.github.io/nlp_course/text_classification.html

# CNN intuition



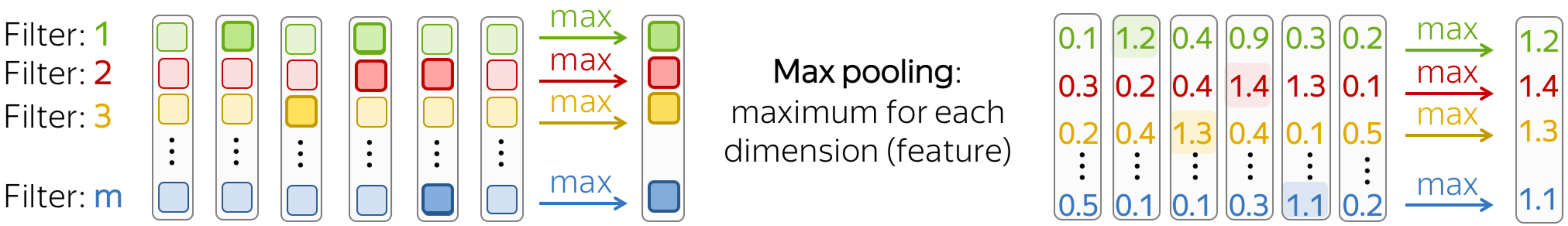

Source: https://lena-voita.github.io/nlp_course/text_classification.html

# CNN intuition



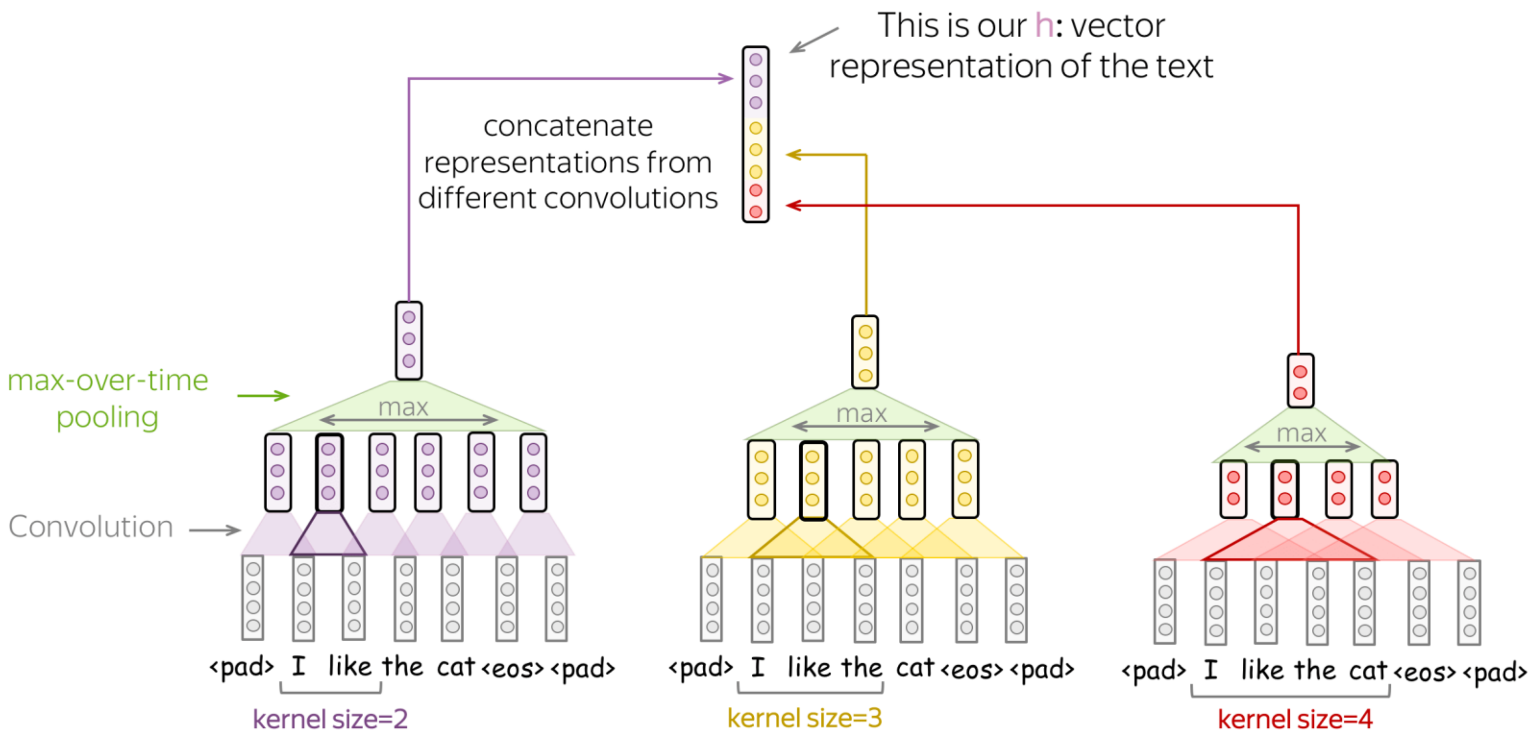

Source: https://lena-voita.github.io/nlp_course/text_classification.html

# Pooling




Source: https://lena-voita.github.io/nlp_course/text_classification.html

# Larger kernel size for a broader context

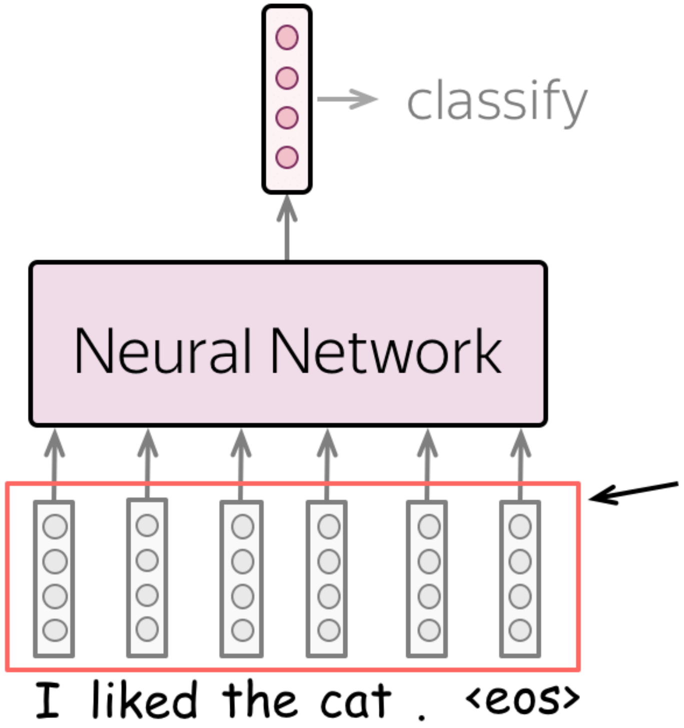



Source: https://lena-voita.github.io/nlp_course/text_classification.html

# Table of Content

- General view on classification with text
- Pre-deep learning approaches
- Deep learning approaches
- **Practical Tips**

# Larger kernel size for a broader context




**Input word embeddings:**

- Train from scratch
- Take pretrained (Word2Vec, GloVe)
- **Initialize with pretrained, then fine-tune**

Source: https://lena-voita.github.io/nlp_course/text_classification.html

# Data augmentation

The movie about cats was absolutely great, and the cats were cute.

pick several words randomly

replace with UNK

The movie UNK cats was absolutely UNK, and the UNK were cute.

replace with random words

The movie mejorate cats was absolutely fellows, and the mak were cute.

Source: https://lena-voita.github.io/nlp_course/text_classification.html

HSE UNIVERSITY • EST 1992 •