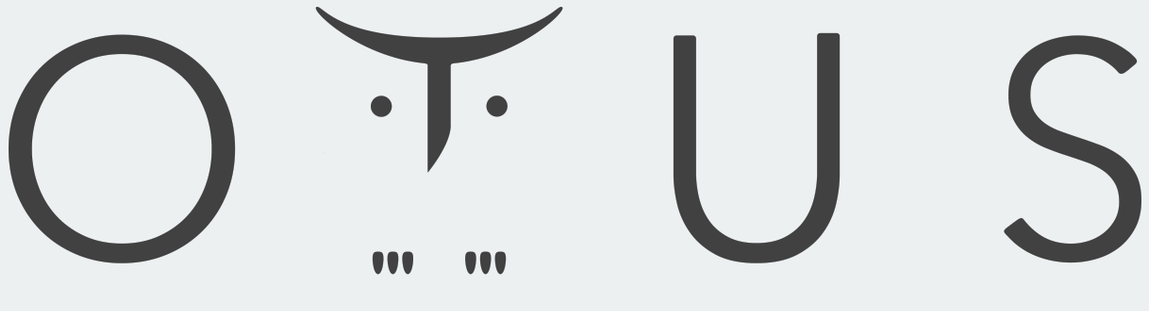
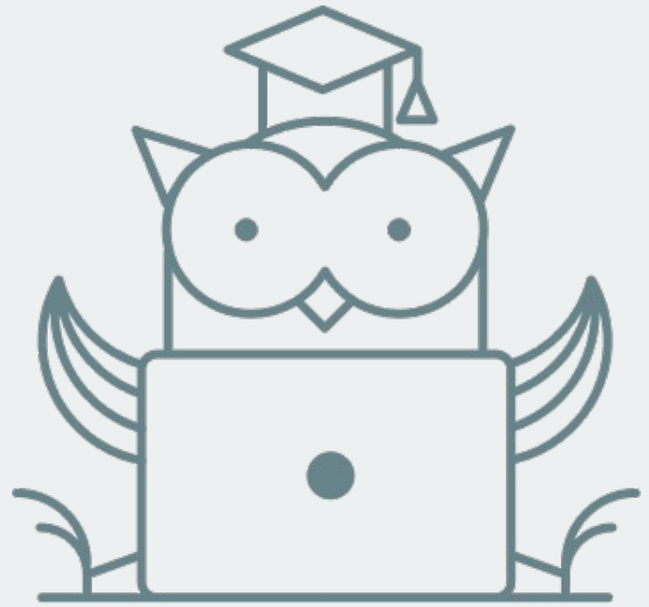


ОНЛАЙН-ОБРАЗОВАНИЕ

NLP – Transfer learning

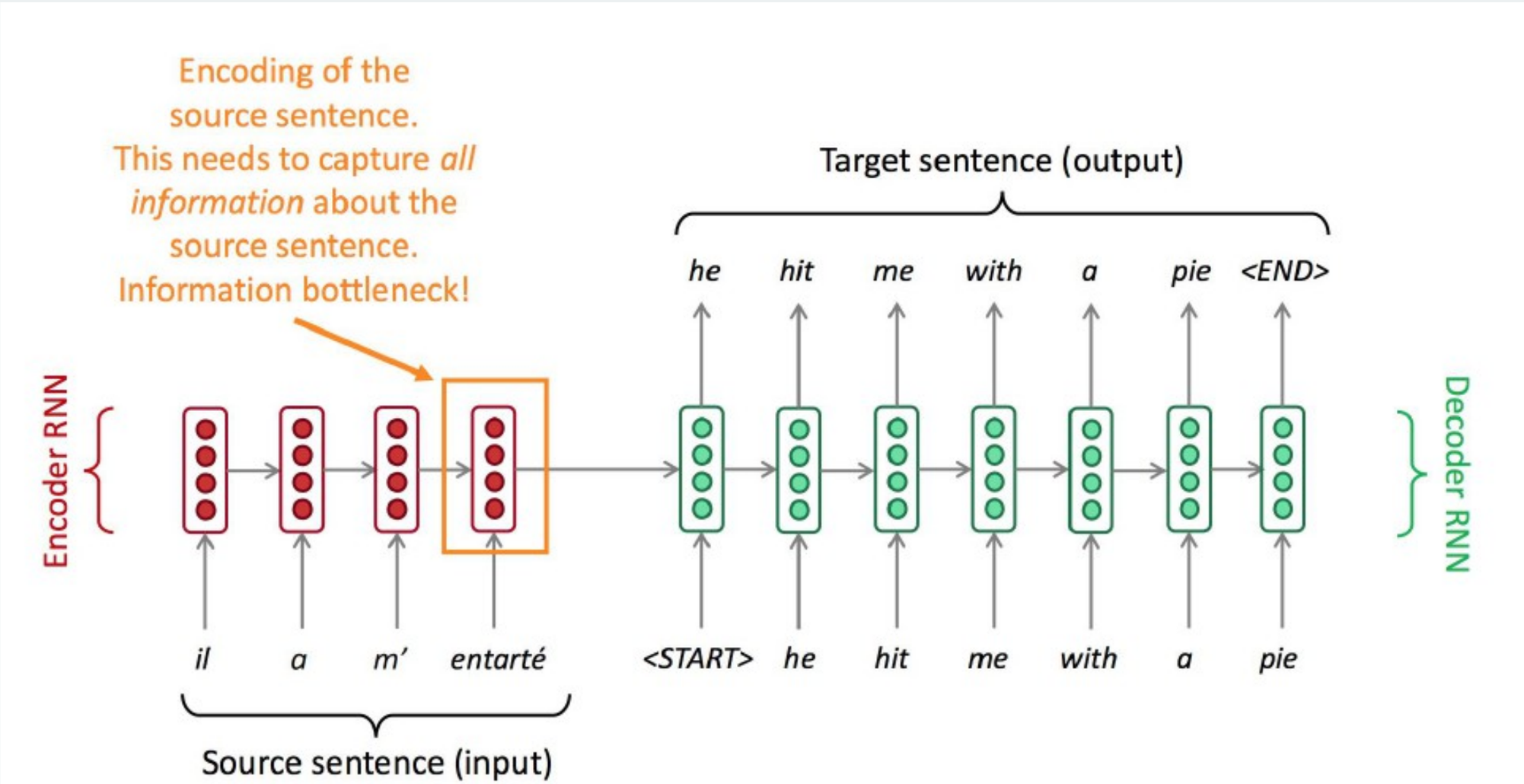
Артур Кадурин
Преподаватель



План на сегодня

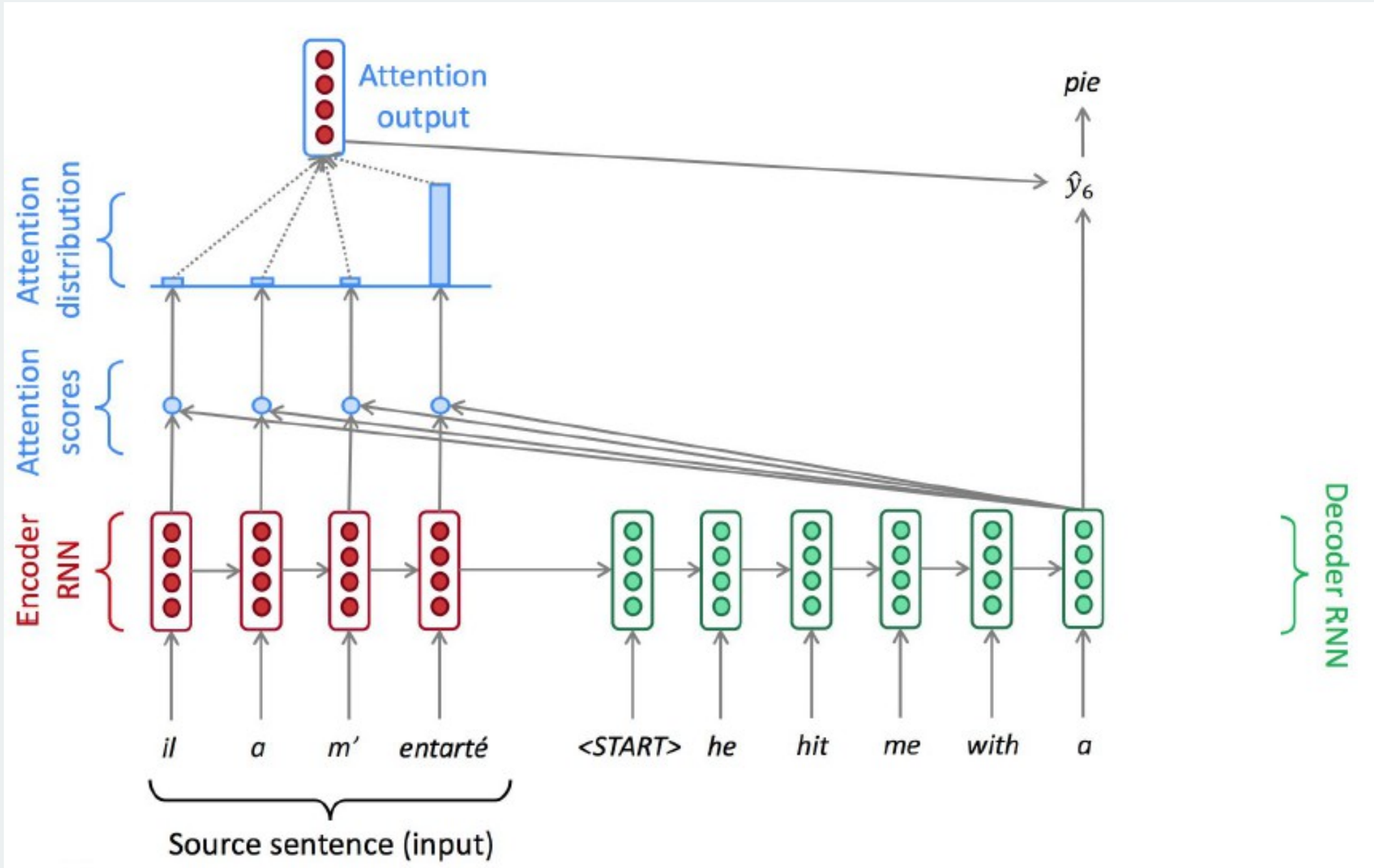
1. Повторение **Attention**
2. **Transformer (OpenAI Transformer)**
3. **BERT**
4. **ELMO**
5. **GPT-2**
6. Практика использования предобученного **BERT**

Attention in seq2seq:



 Артур Кадурин, преподаватель

Attention in seq2seq:



 Артур Кадурин, преподаватель

OTUS

BLEU score

> **bilingual evaluation understudy**

Коэффициент "немногословности" (brevity) $B = \begin{cases} e^{\left(1-\frac{|ref|}{|hyp|}\right)}, & \text{if } |ref| > |hyp| \\ 1, & \text{otherwise} \end{cases}$

$$p_n = \frac{\sum_{n-grap \in hypothesis} count-clip(n-gram)}{\sum_{n-grap \in hypothesis} count(n-gram)}$$

$$BLEU = B \cdot exp\left[\frac{1}{N}\sum_{n=1}^{N} p_n\right]$$

Артур Кадурин, преподаватель

Attention in seq2seq:

We have encoder hidden states $h_1, \ldots, h_N \in \mathbb{R}^h$

On timestep *t*, we have decoder hidden state $s_t \in \mathbb{R}^h$

We get the attention scores e^t for this step:

$$e^t = [s_t^T h_1, \ldots, s_t^T h_N] \in \mathbb{R}^N$$

We take softmax to get the attention distribution α^t for this step (this is a probability distribution and sums to 1)

$$\alpha^t = \text{softmax}(e^t) \in \mathbb{R}^N$$

We use α^t to take a weighted sum of the encoder hidden states to get the attention output a_t

$$a_t = \sum_{i=1}^{N} \alpha_i^t h_i \in \mathbb{R}^h$$

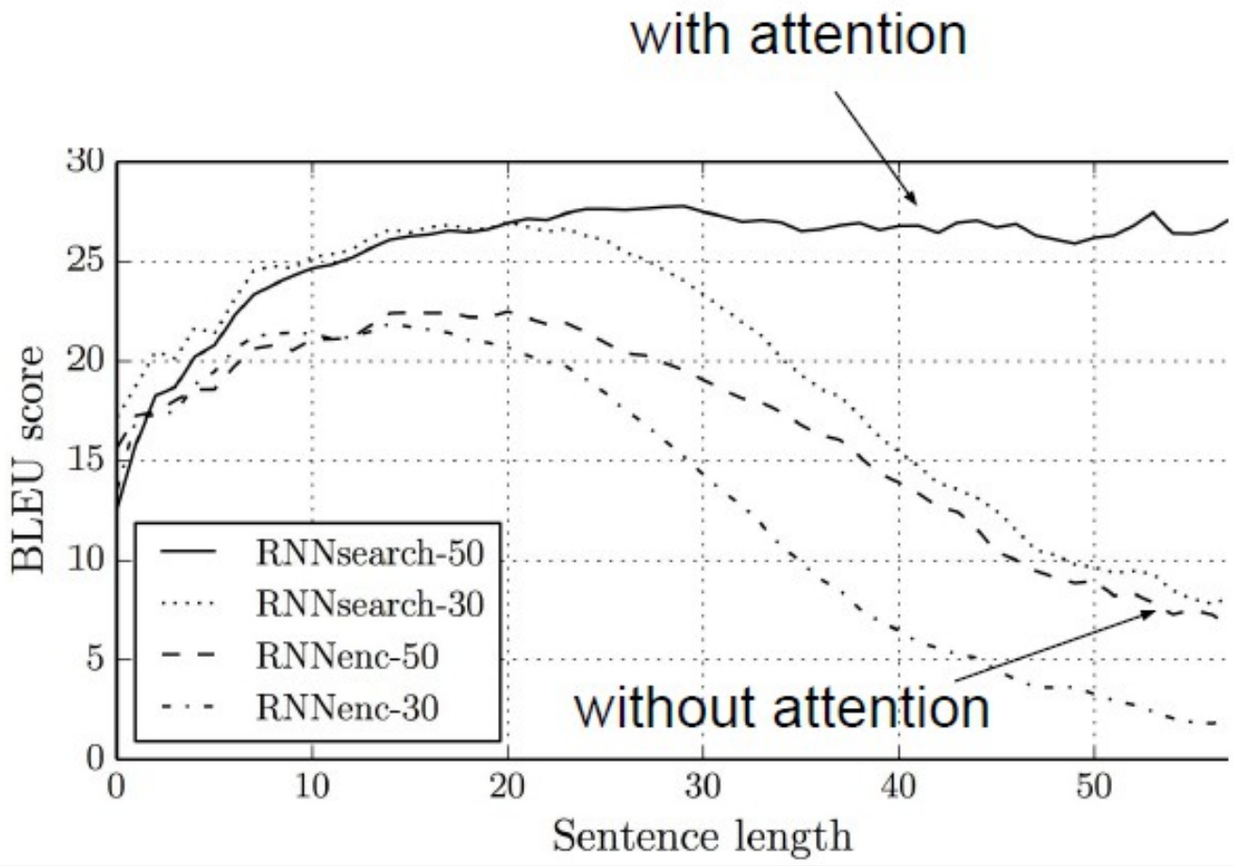
Finally we concatenate the attention output a_t with the decoder hidden state s_t and proceed as in the non-attention seq2seq model

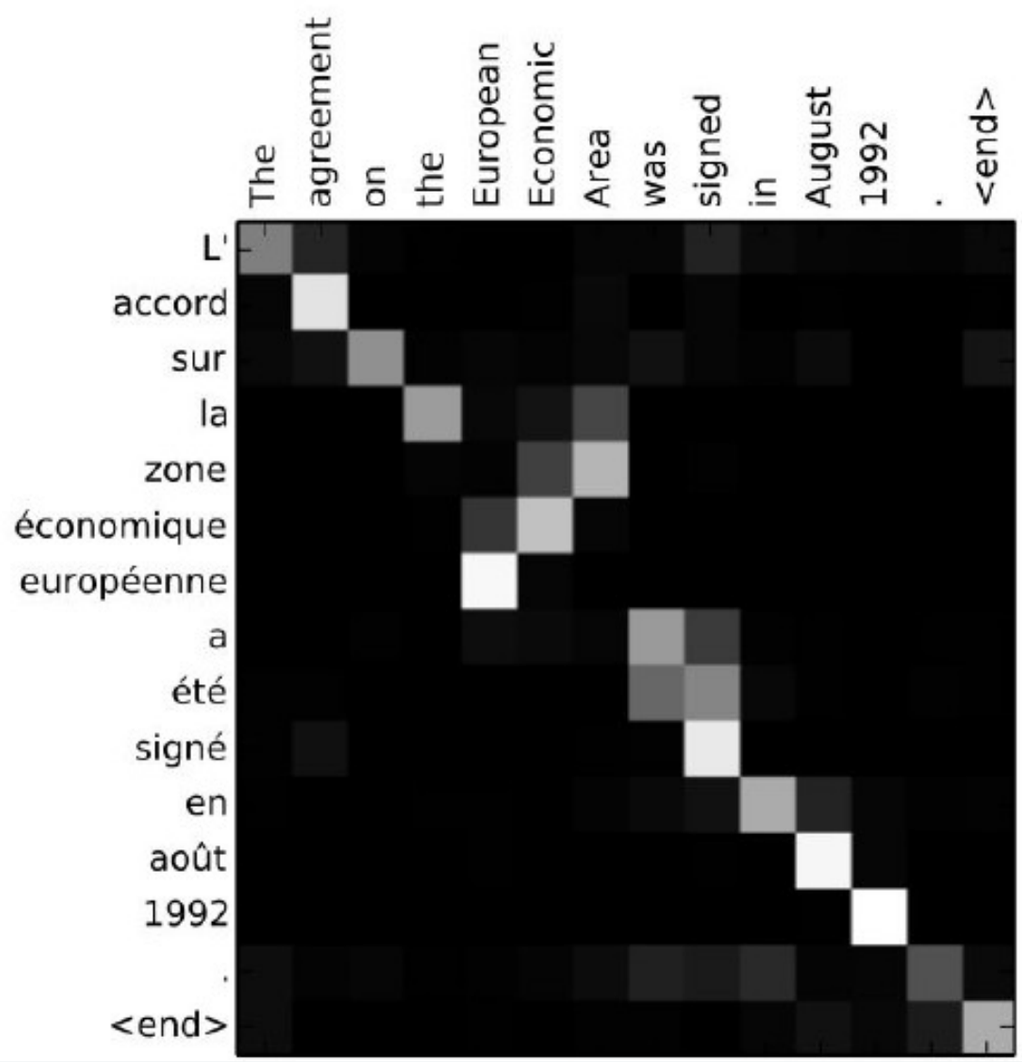
$$[a_t; s_t] \in \mathbb{R}^{2h}$$

Артур Кадурин, преподаватель

Attention:

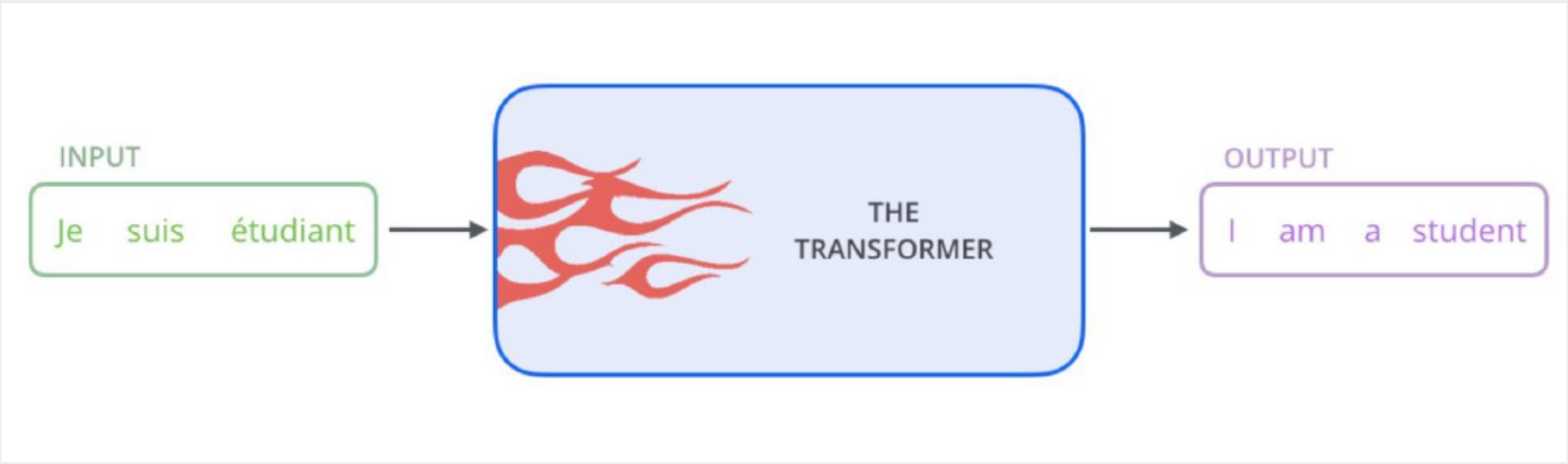
- “Free” word alignment
- Better results on long sequences





Артур Кадурин, преподаватель

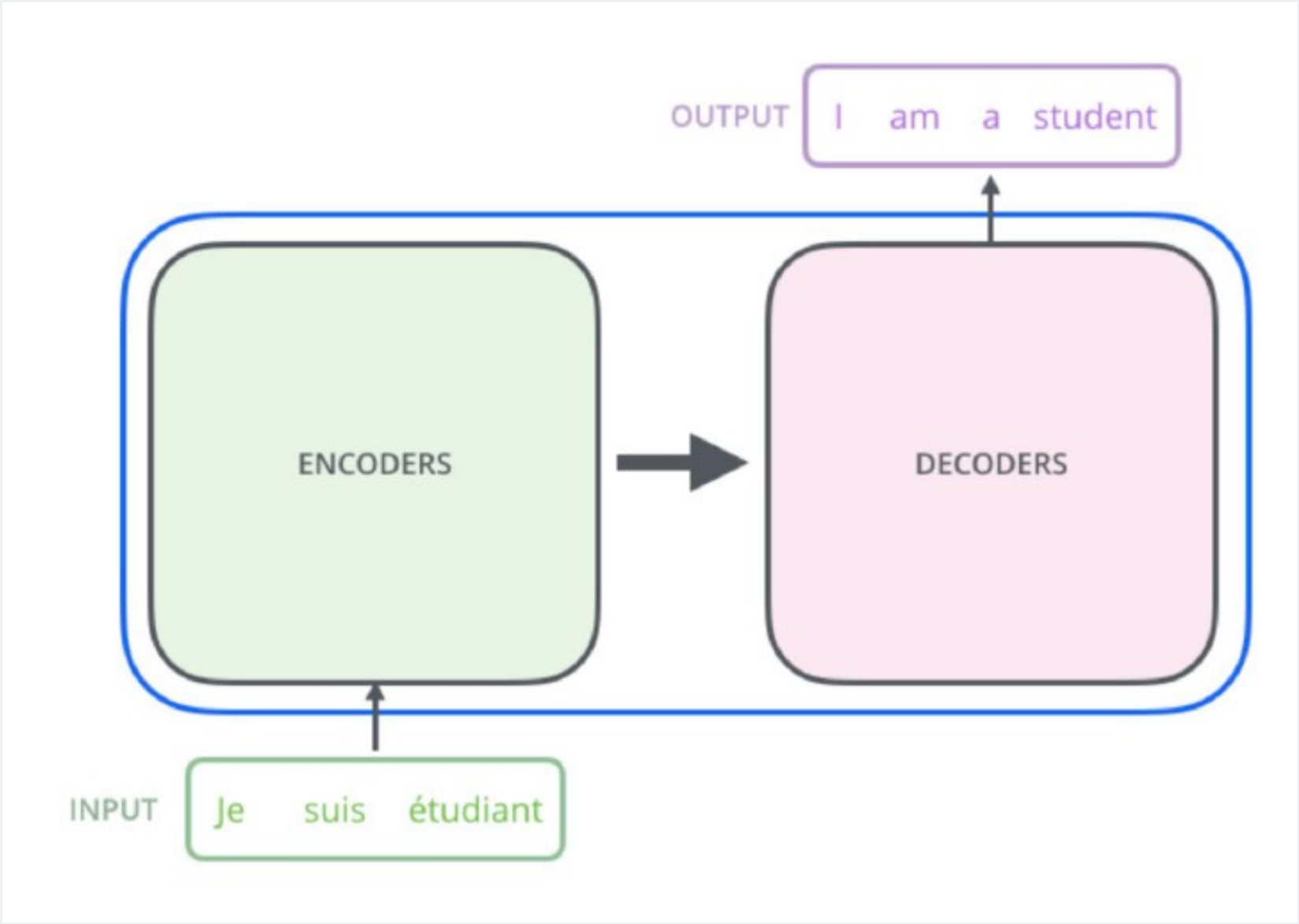
Transformer



Иллюстрации взяты отсюда: http://jalammar.github.io/illustrated-transformer/

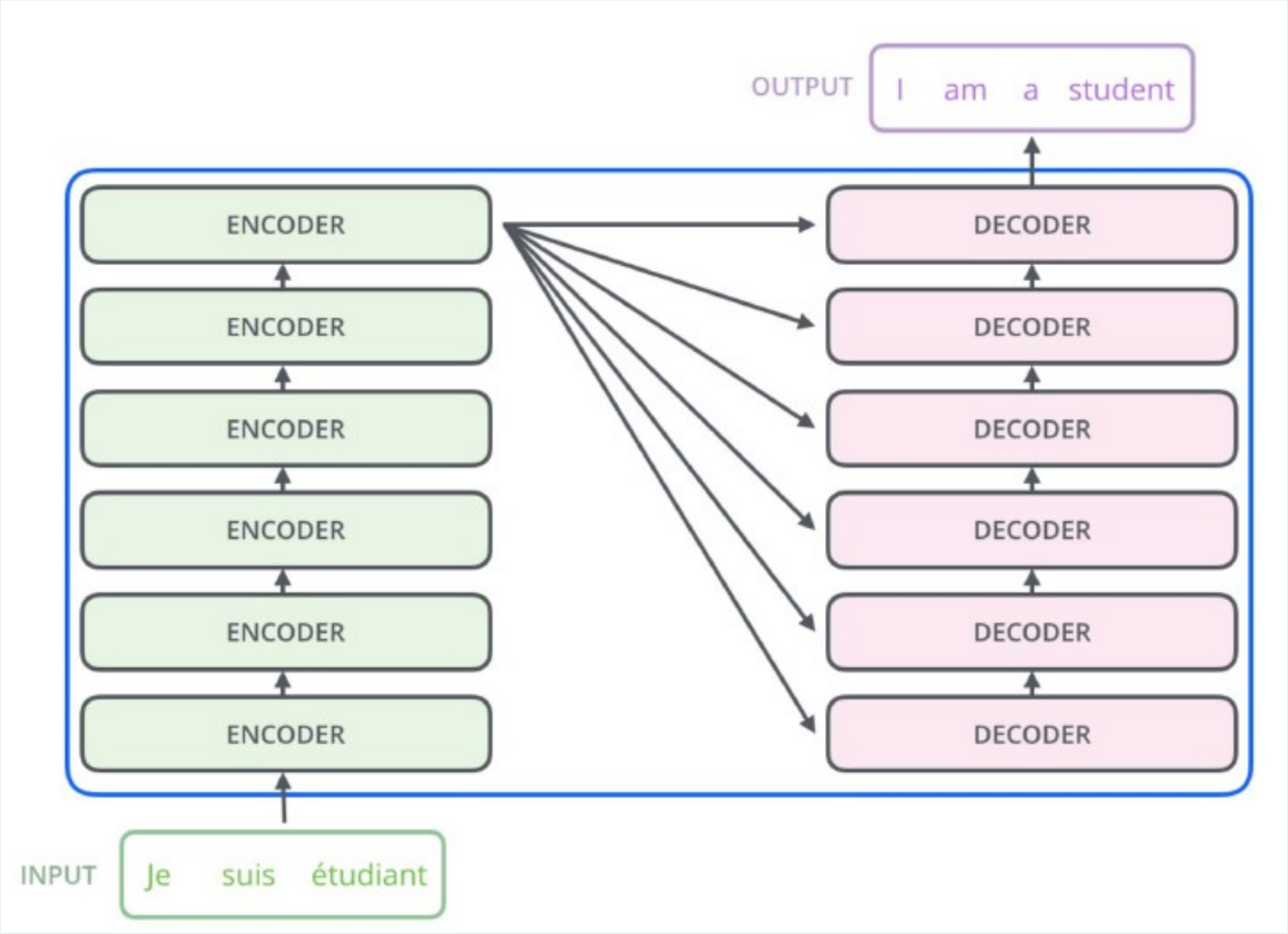
Артур Кадурин, преподаватель

Transformer



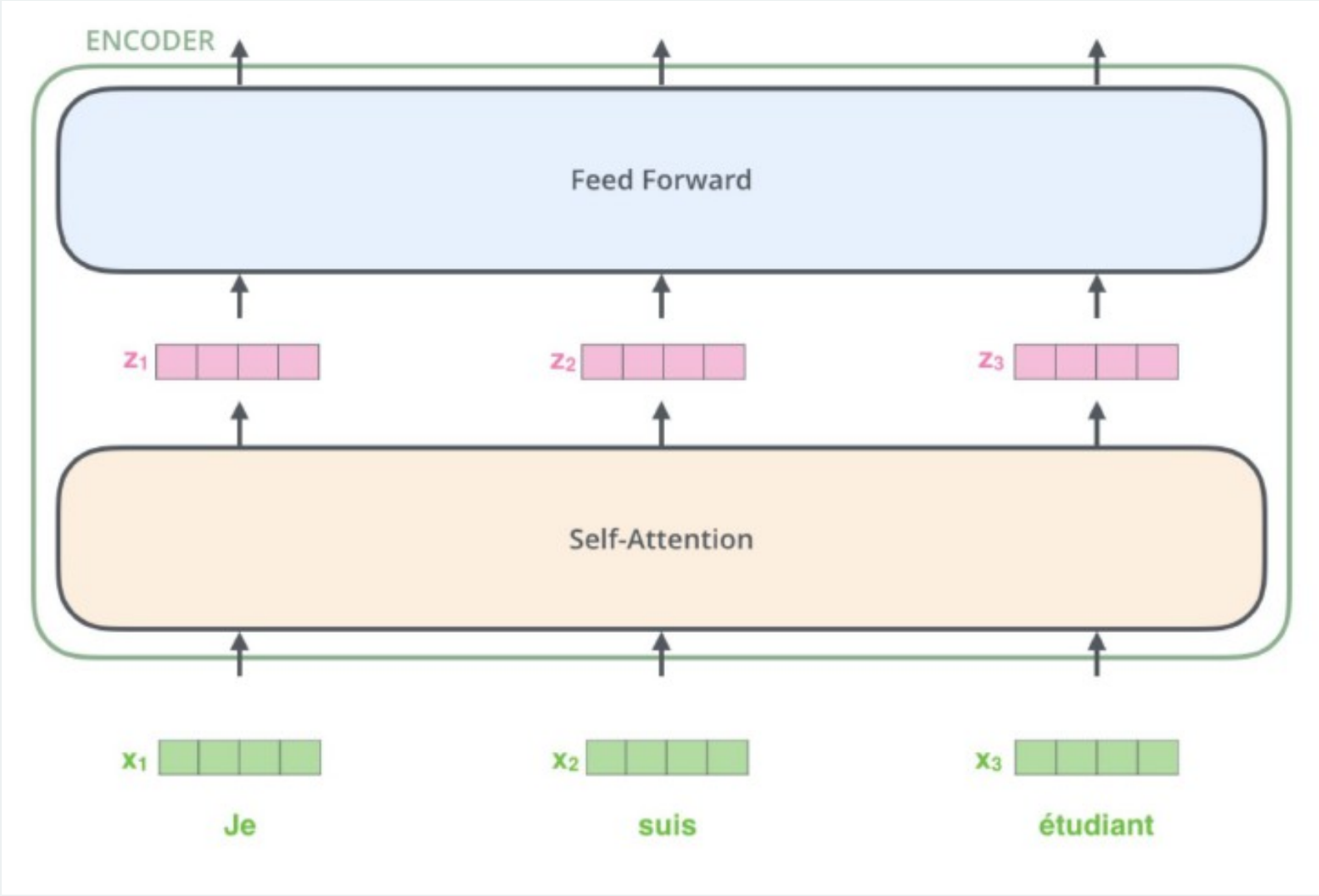
1 **Артур Кадурин, преподаватель**

Transformer



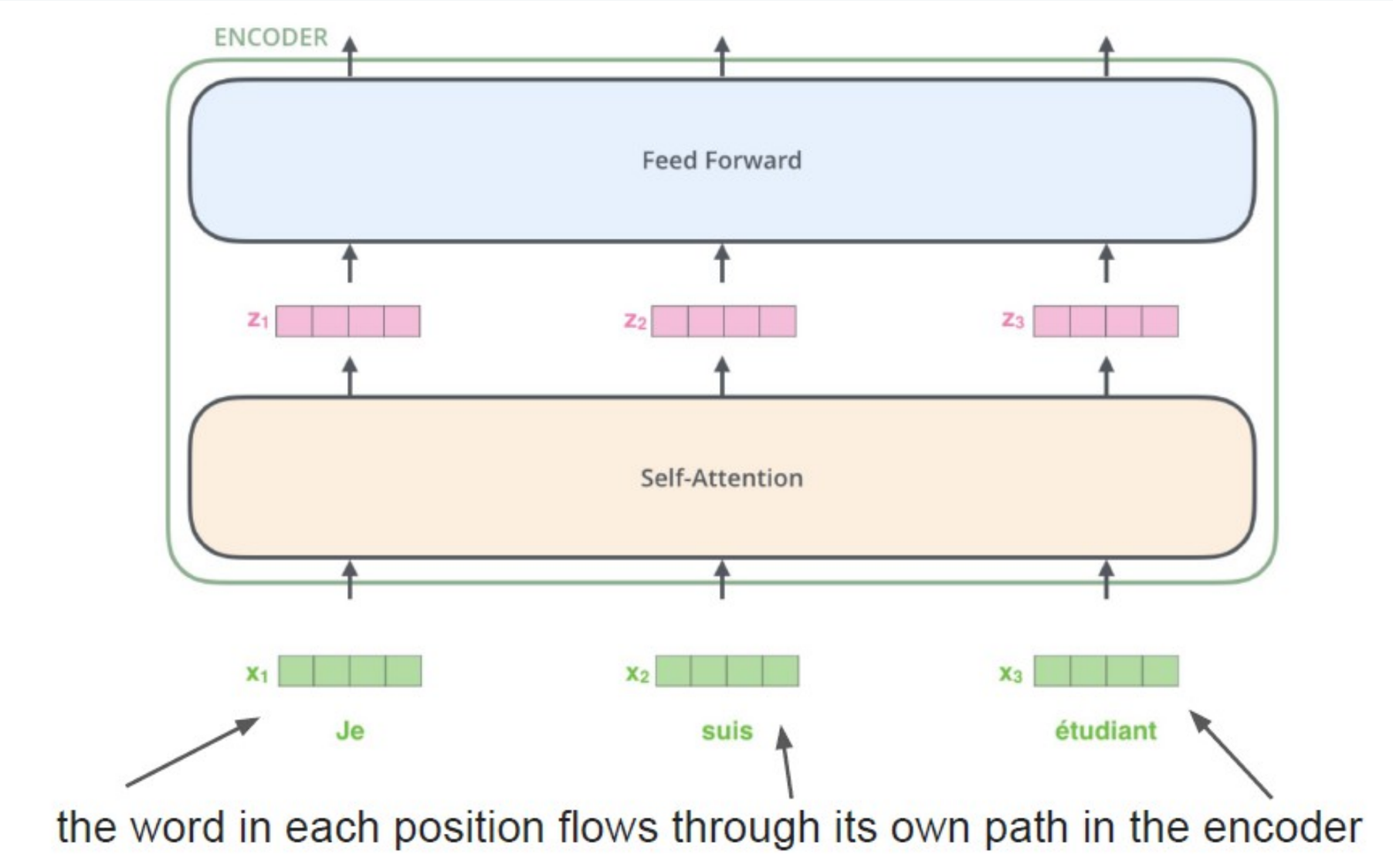
1 **Артур Кадурин, преподаватель**

Transformer



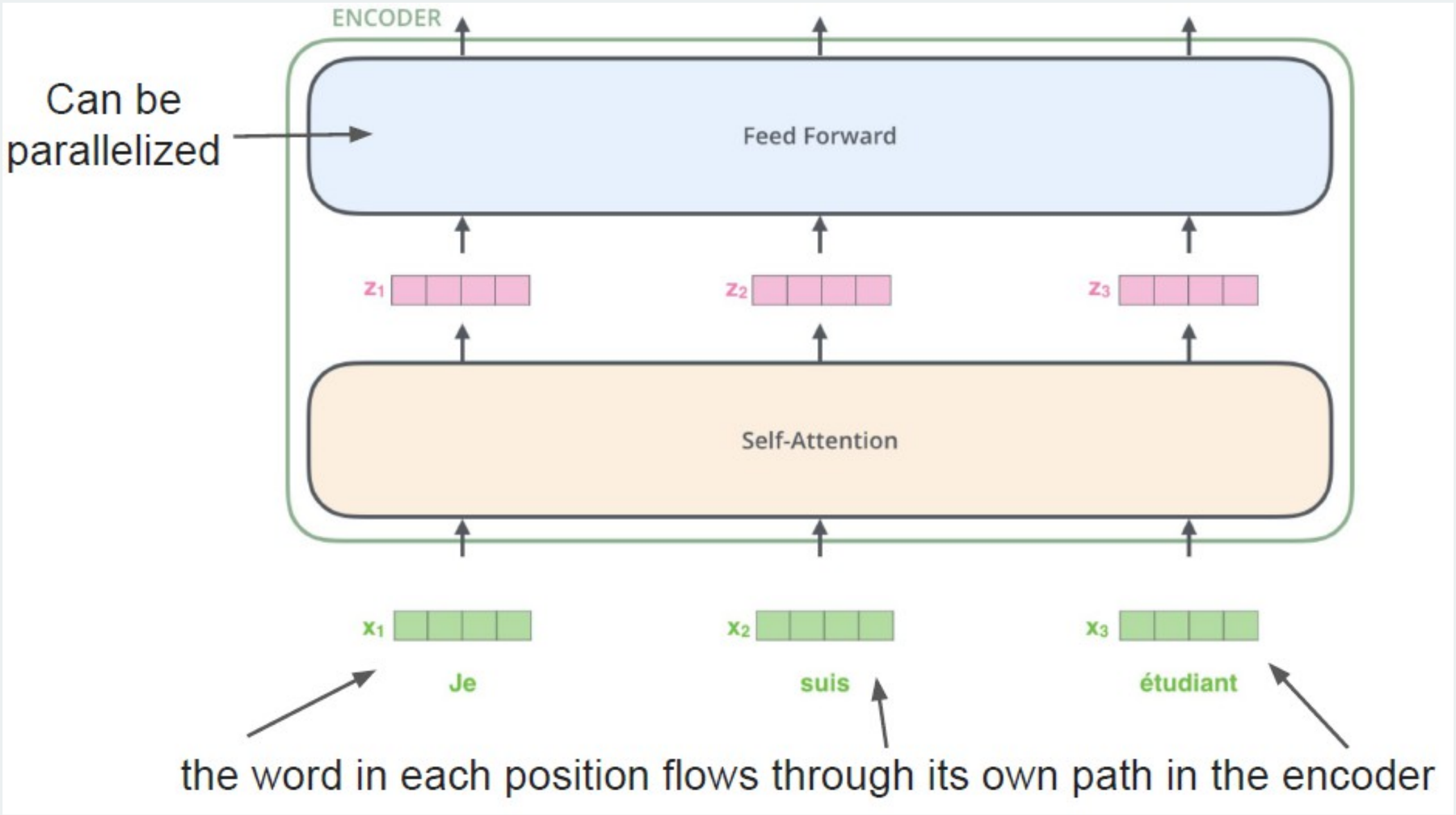
Артур Кадурин, преподаватель

Transformer



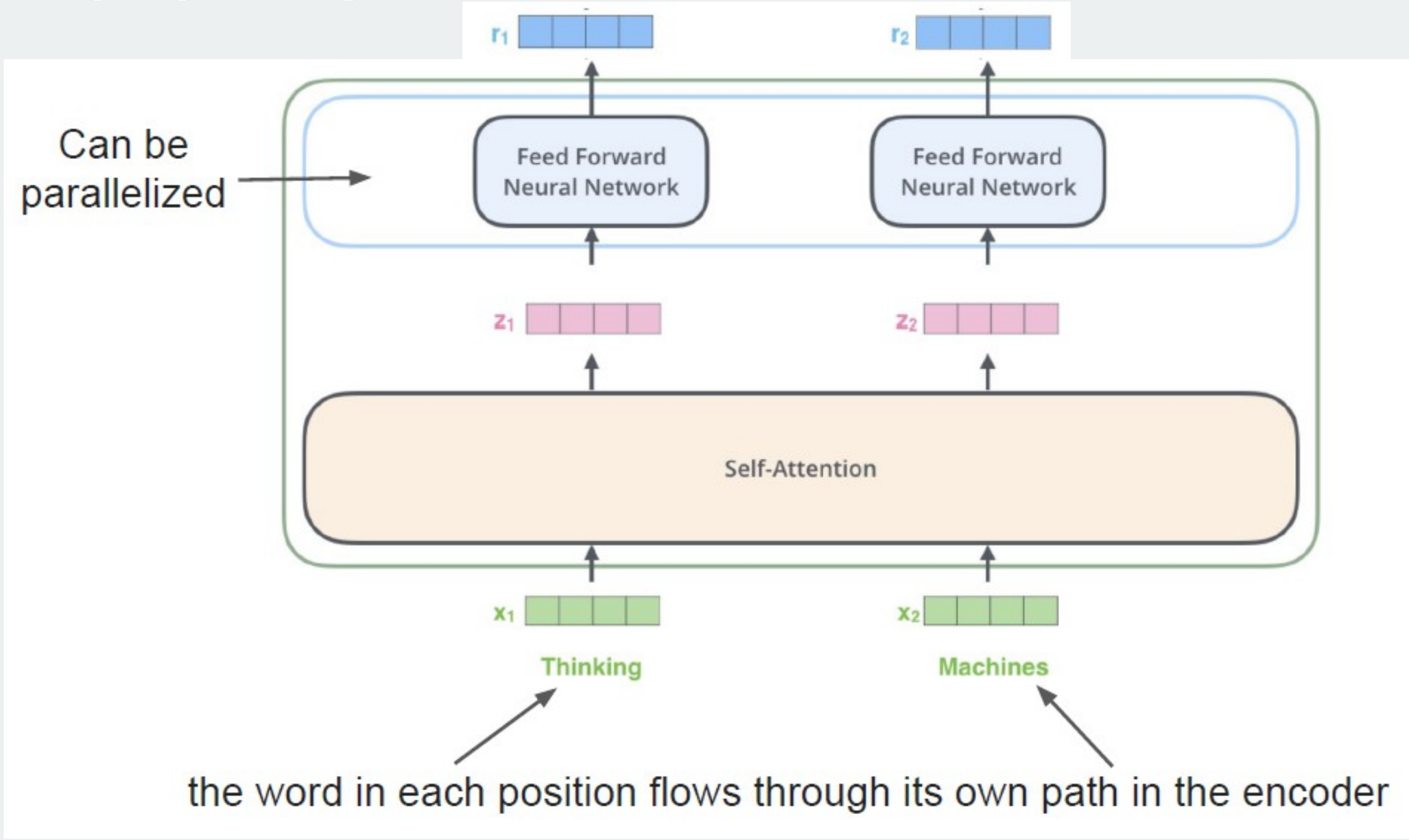
1 **Артур Кадурин, преподаватель**

Transformer



1 **Артур Кадурин, преподаватель**

Transformer



1 **Артур Кадурин, преподаватель**

Transformer

”The animal didn't cross the street because it was too tired”

- What does “it” in this sentence refer to?

Артур Кадурин, преподаватель

Transformer

”The animal didn't cross the street because it was too tired”

- What does “it” in this sentence refer to?
- We want self-attention to associate “it” with “animal”

Transformer

"The animal didn't cross the street because it was too tired"

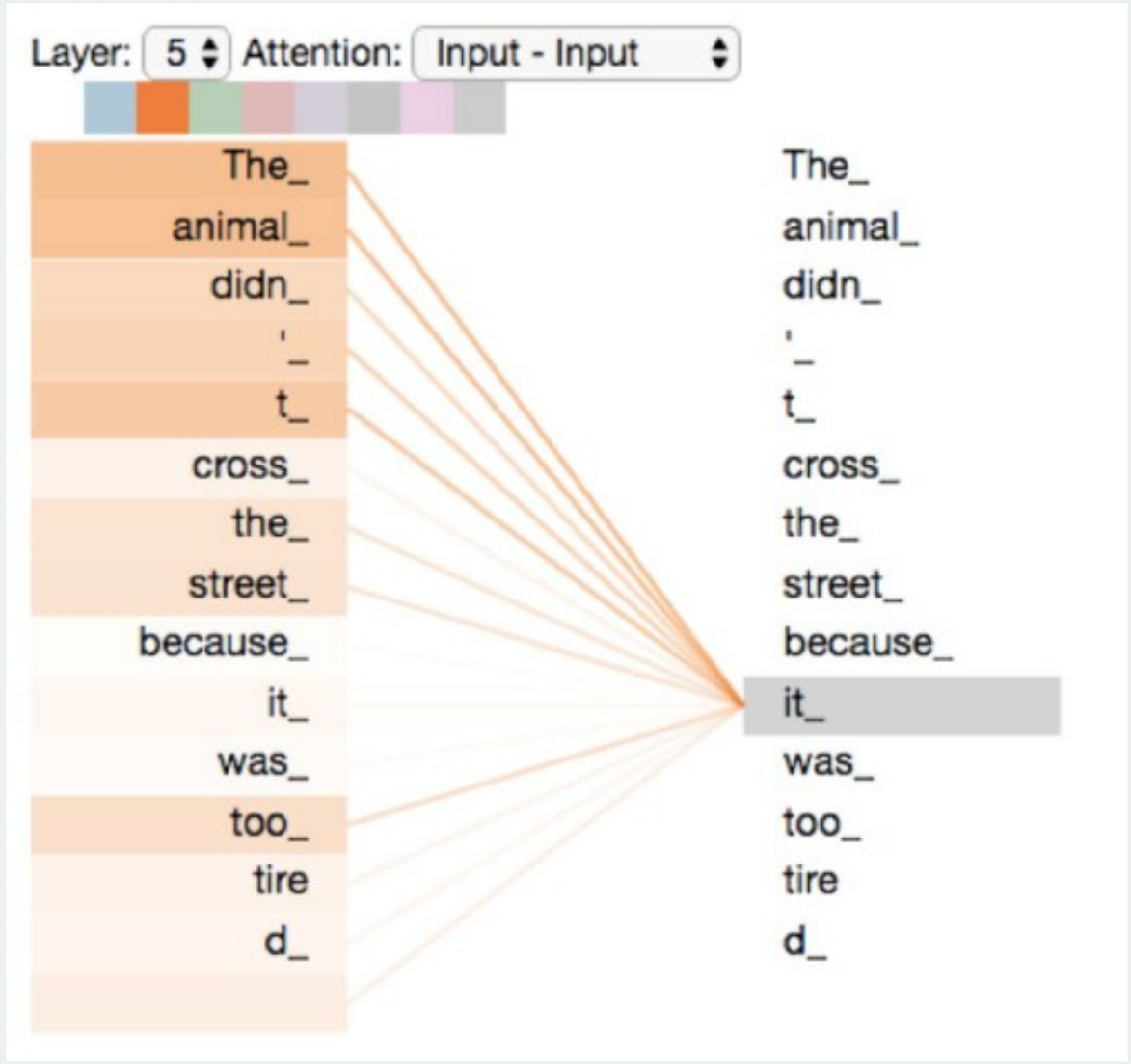
- What does “it” in this sentence refer to?
- We want self-attention to associate “it” with “animal”

- Self-attention is the method the Transformer uses to bake the “understanding” of other relevant words into the one we’re currently processing

Артур Кадурин, преподаватель

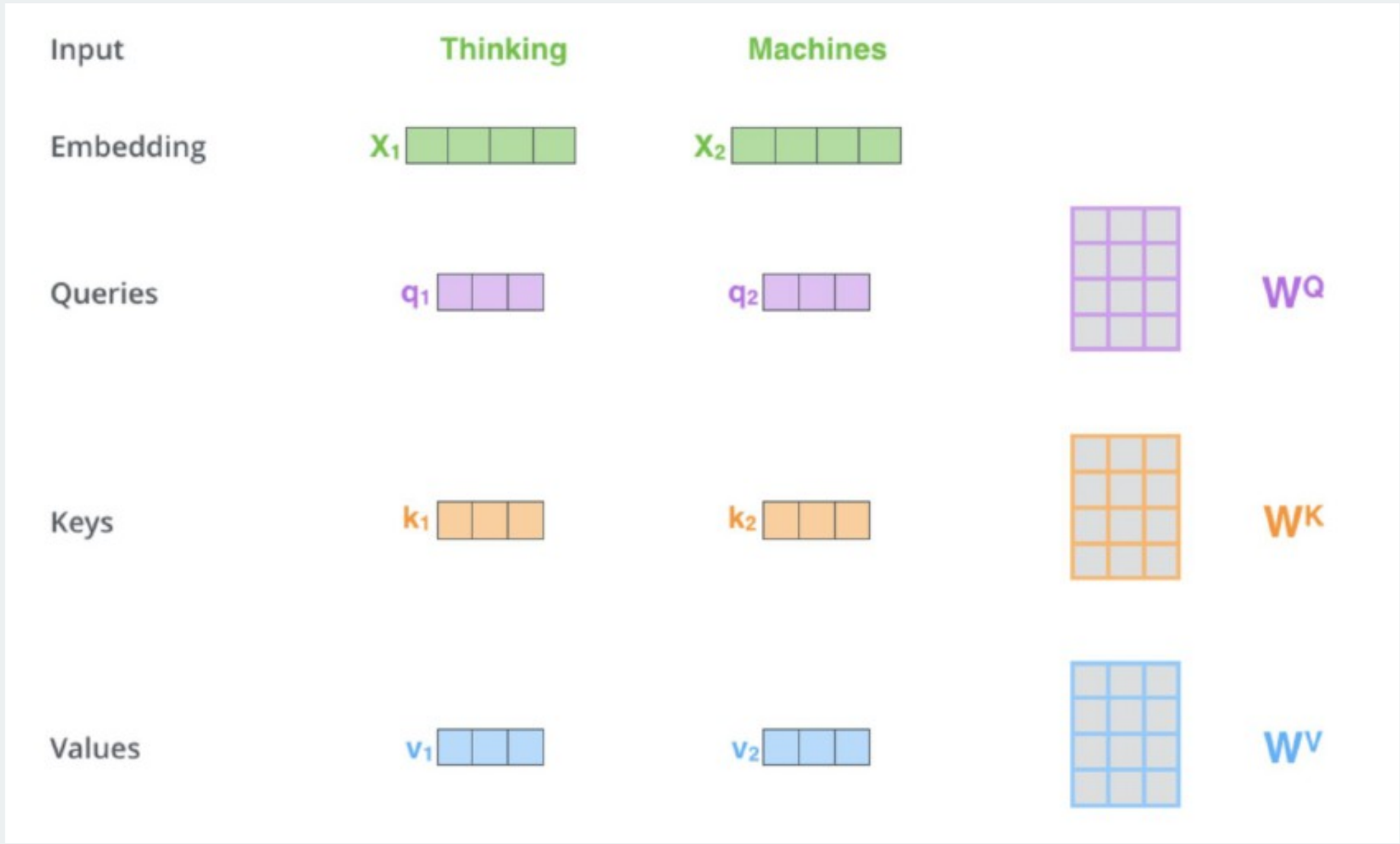
OTUS

Transformer



1 **Артур Кадурин, преподаватель**

Transformer

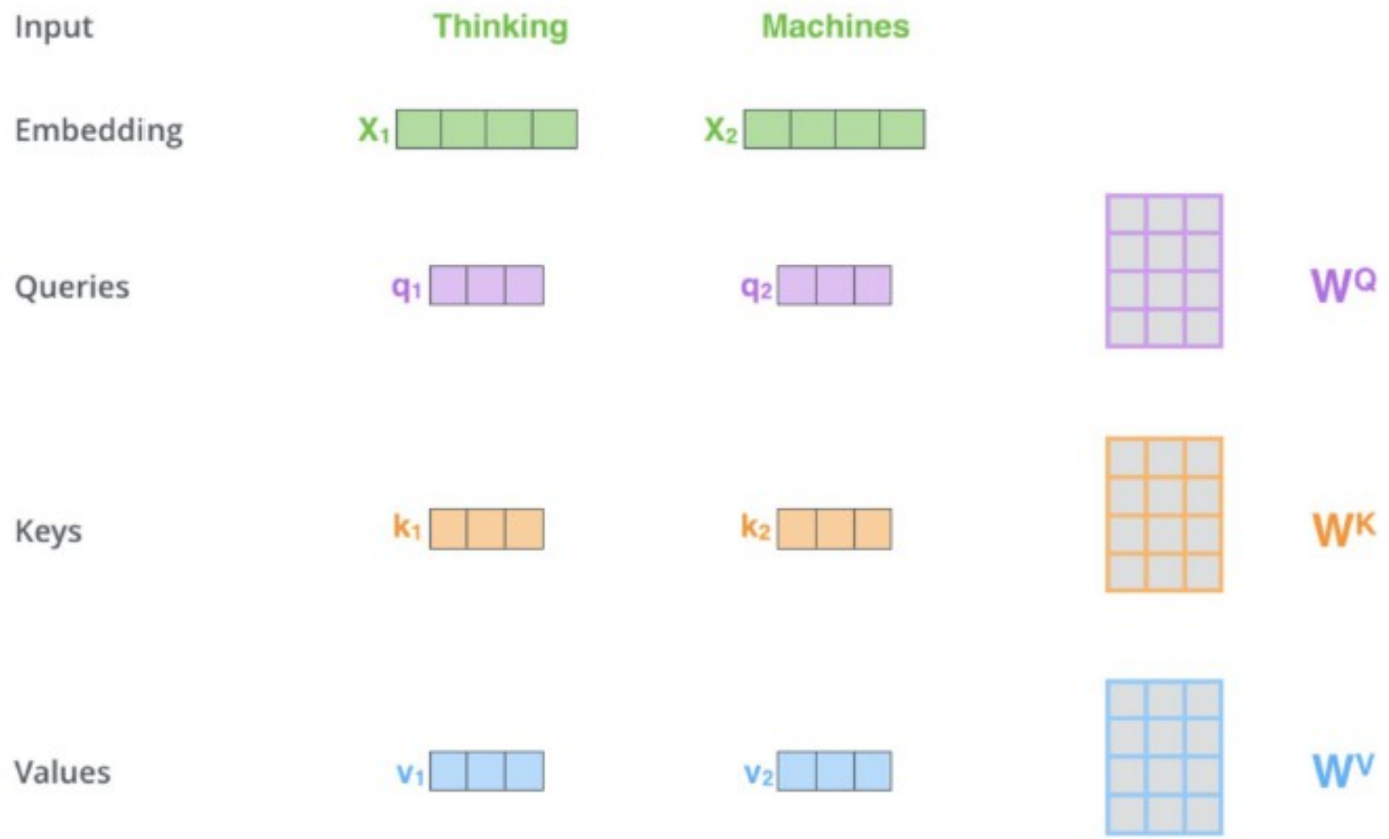


1 **Артур Кадурин, преподаватель**

Transformer

STEP 1:

create 3 vectors (Query, Key, Value) from each of the encoder's input vectors



Артур Кадурин, преподаватель

Transformer

What are the “query”, “key”, and “value” vectors?

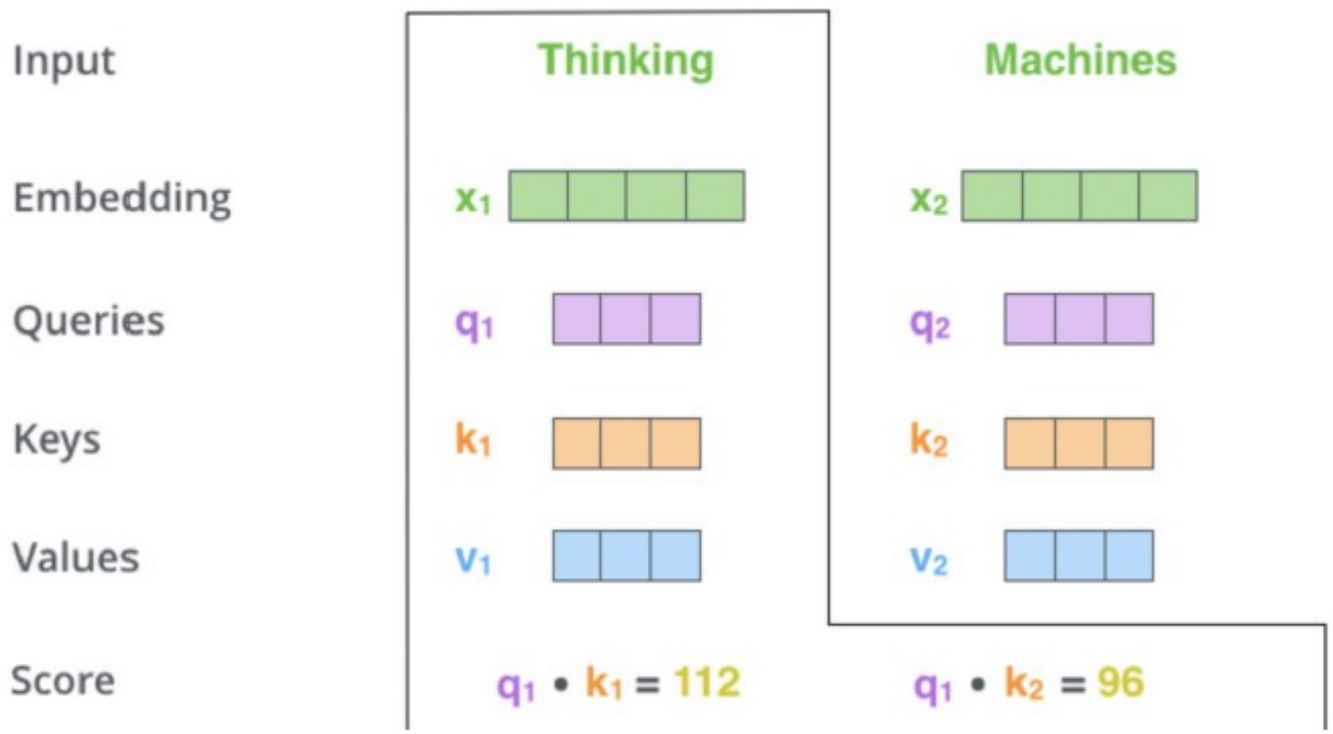
They’re abstractions that are useful for calculating and thinking about attention.

Transformer

STEP 2:

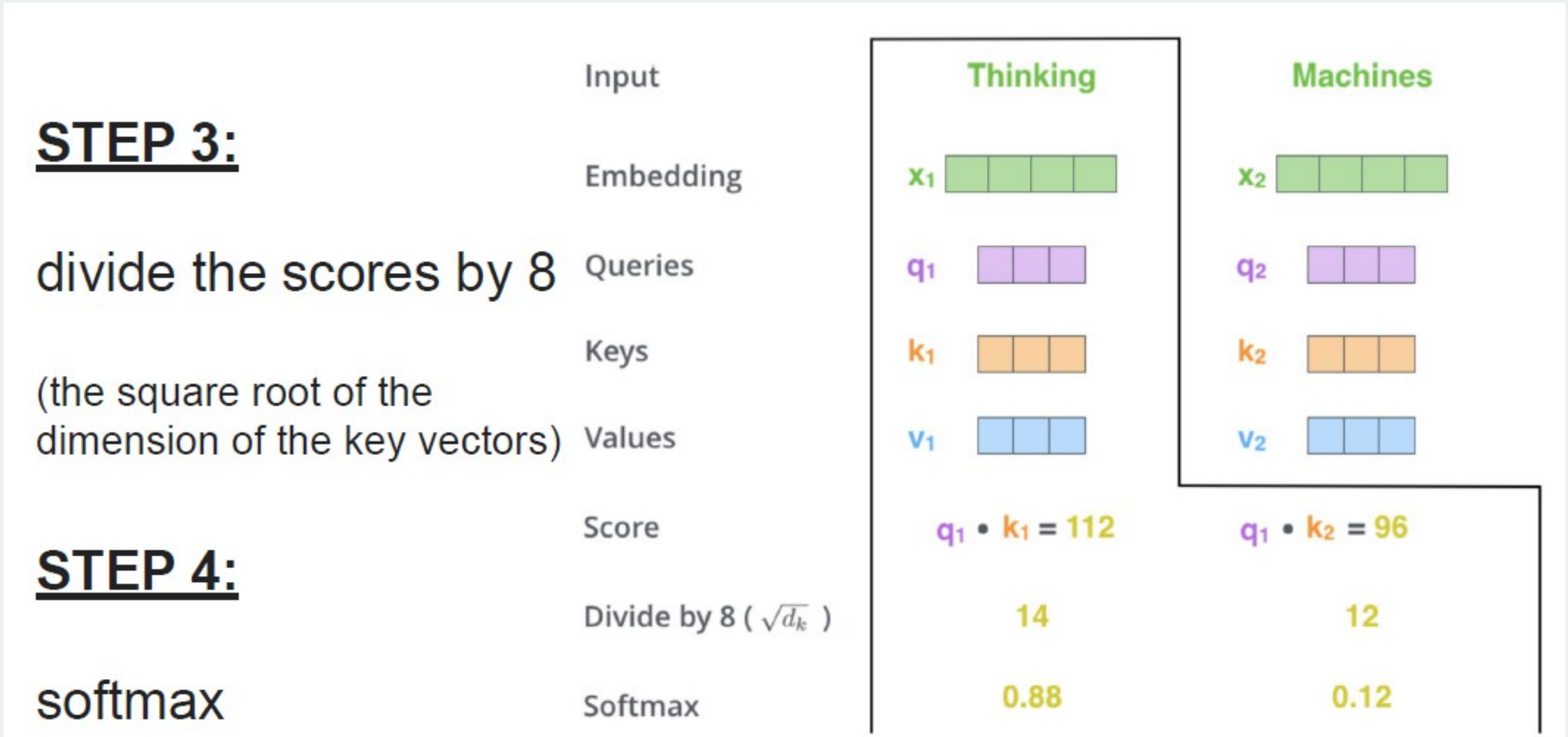
calculate a score

(score each word of the input sentence against the current word)



 Артур Кадурин, преподаватель

Transformer



 Артур Кадурин, преподаватель

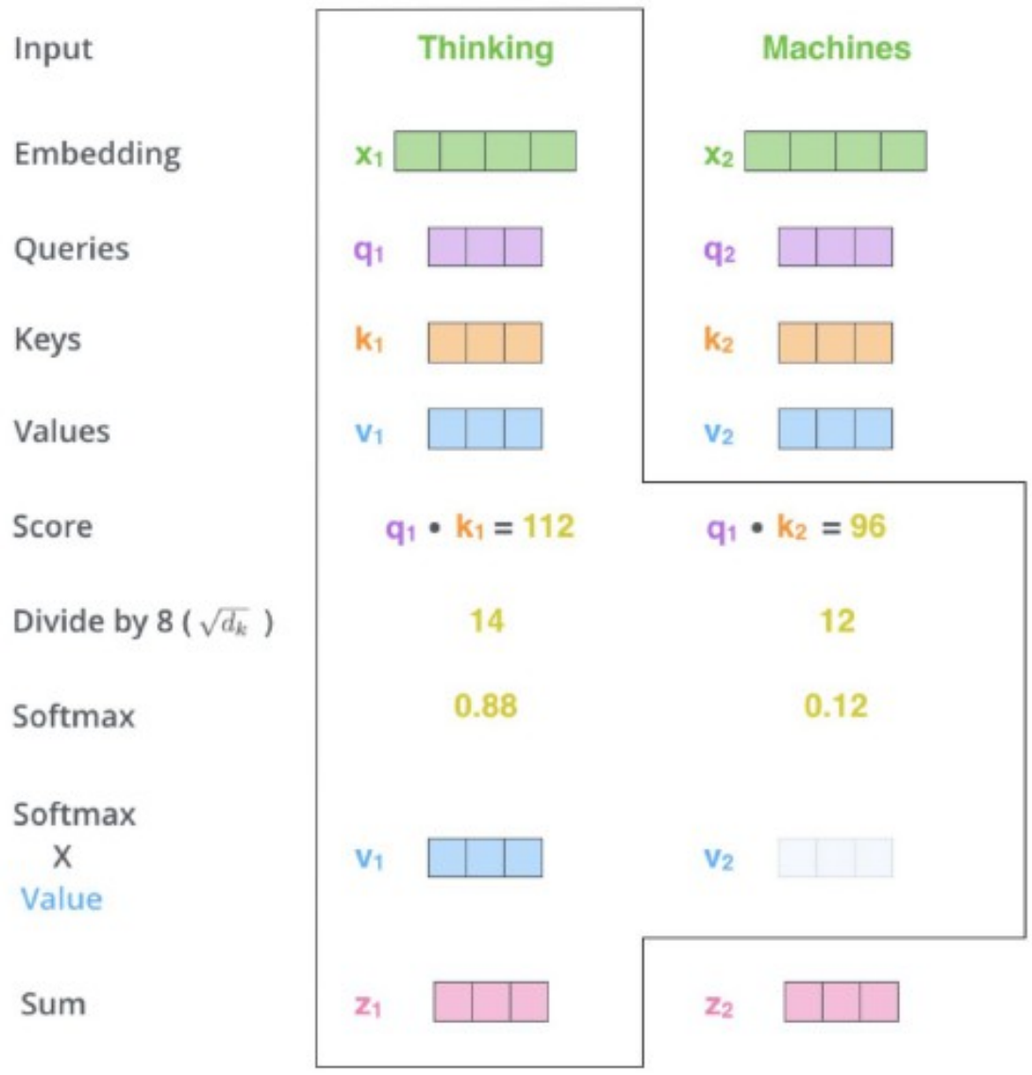
Transformer

STEP 5:

multiply each value vector by the softmax score

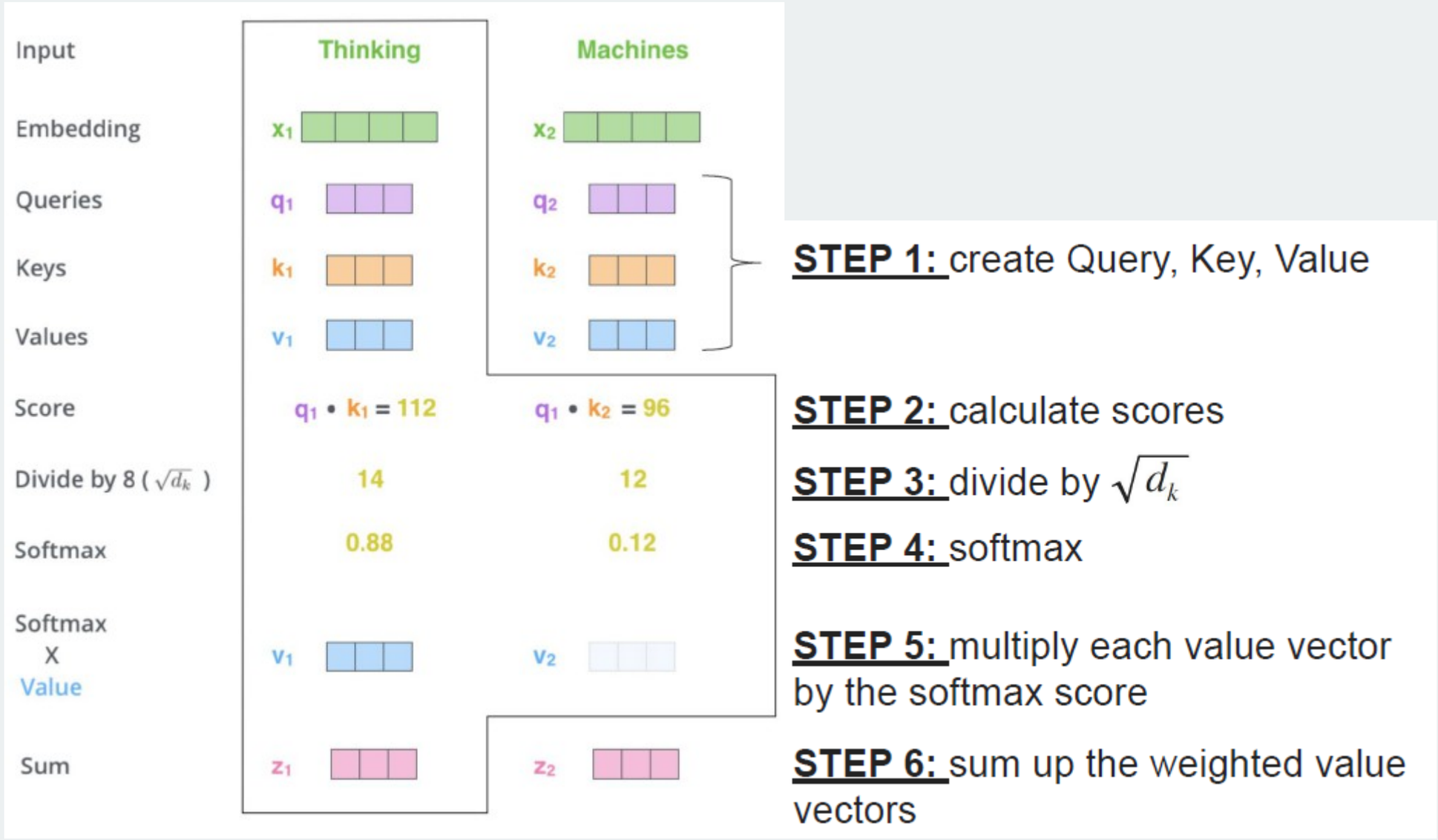
STEP 6:

sum up the weighted value vectors



 Артур Кадурин, преподаватель

Transformer

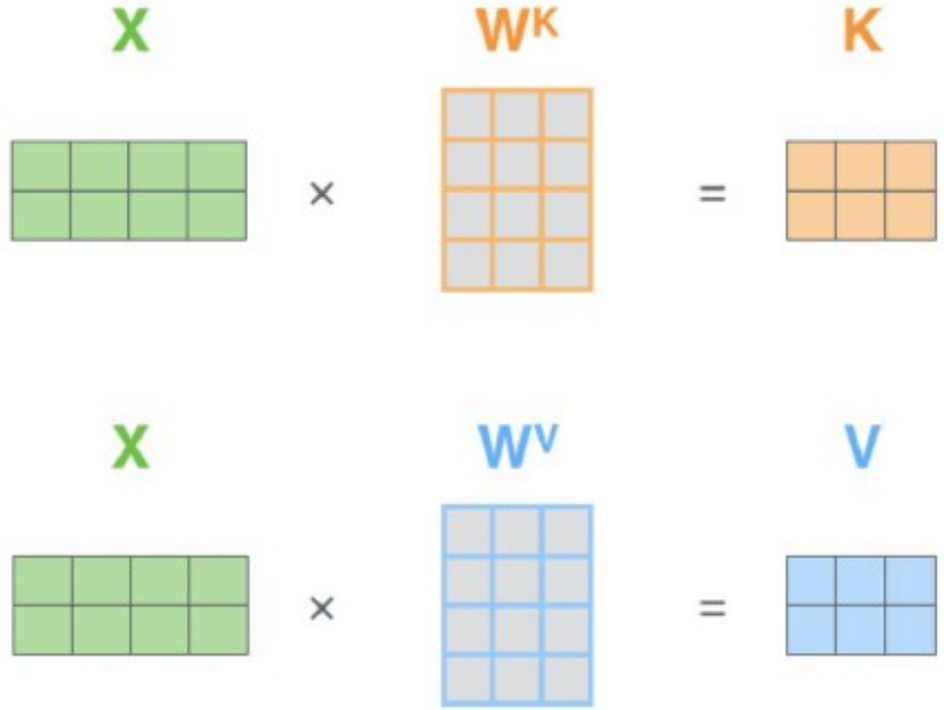


1 **Артур Кадурин, преподаватель**

Transformer

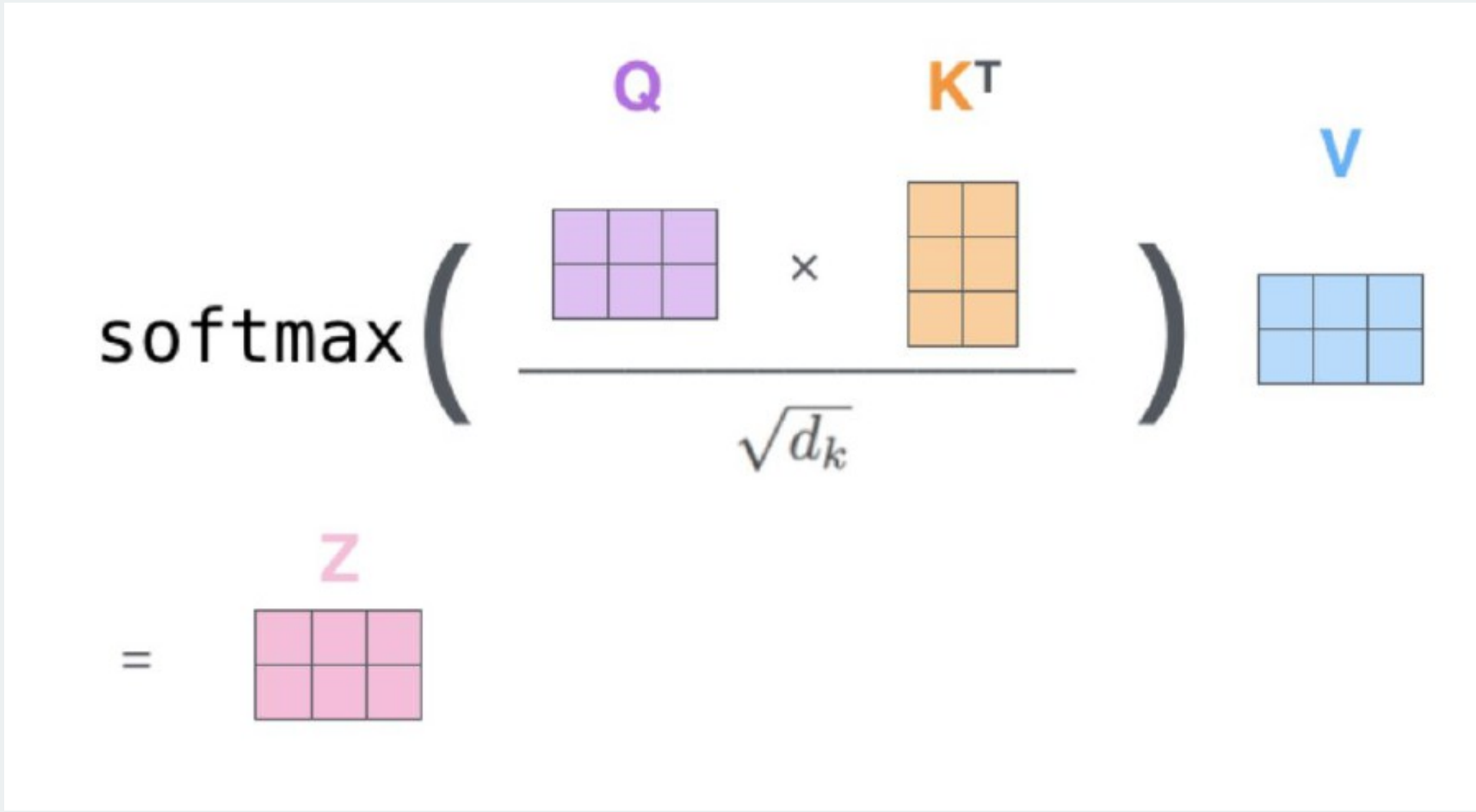
Pack embeddings into matrix **X**

Multiply **X** by weight matrices we've trained (**Wk, Wq, Wv**)



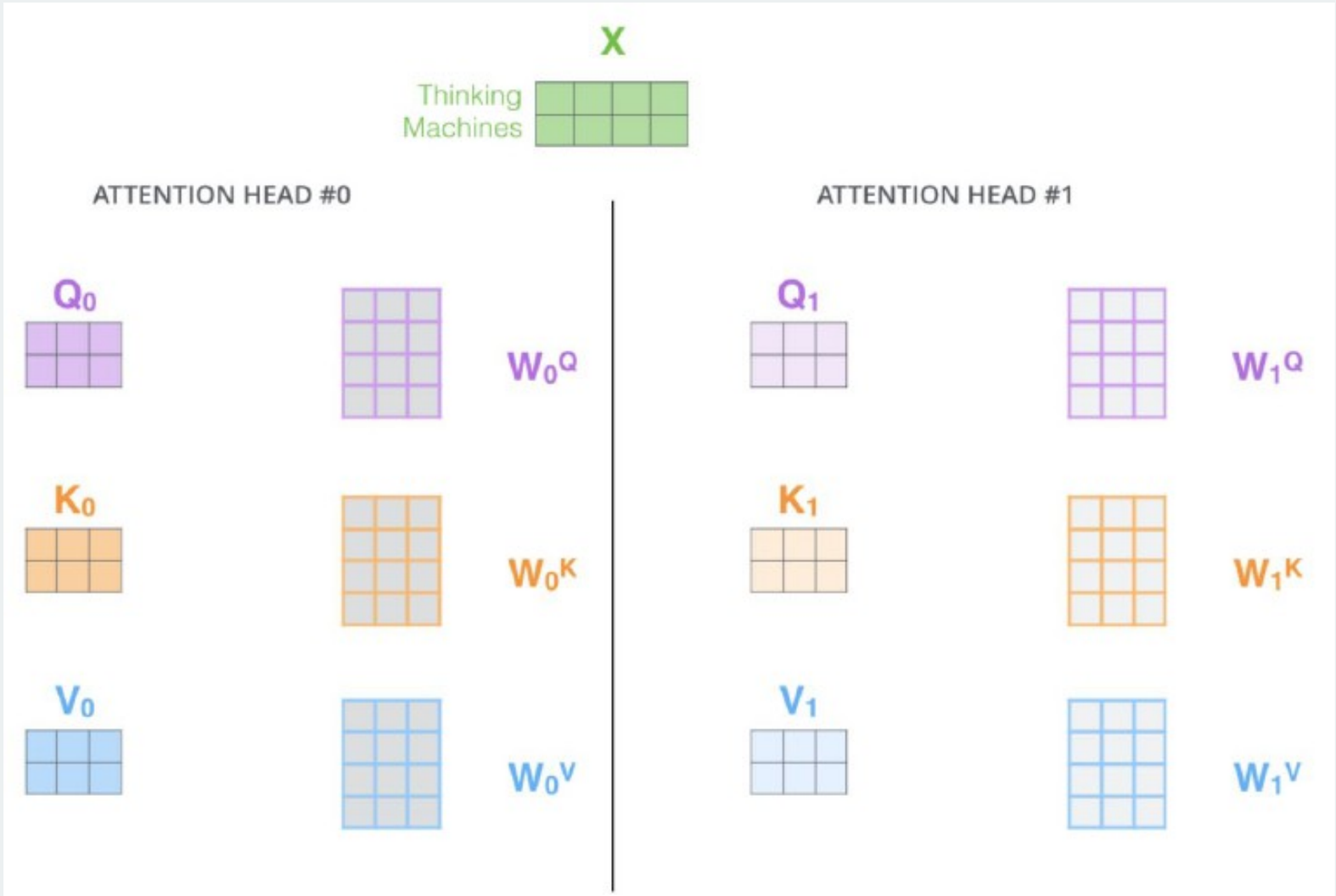
Артур Кадурин, преподаватель

Transformer



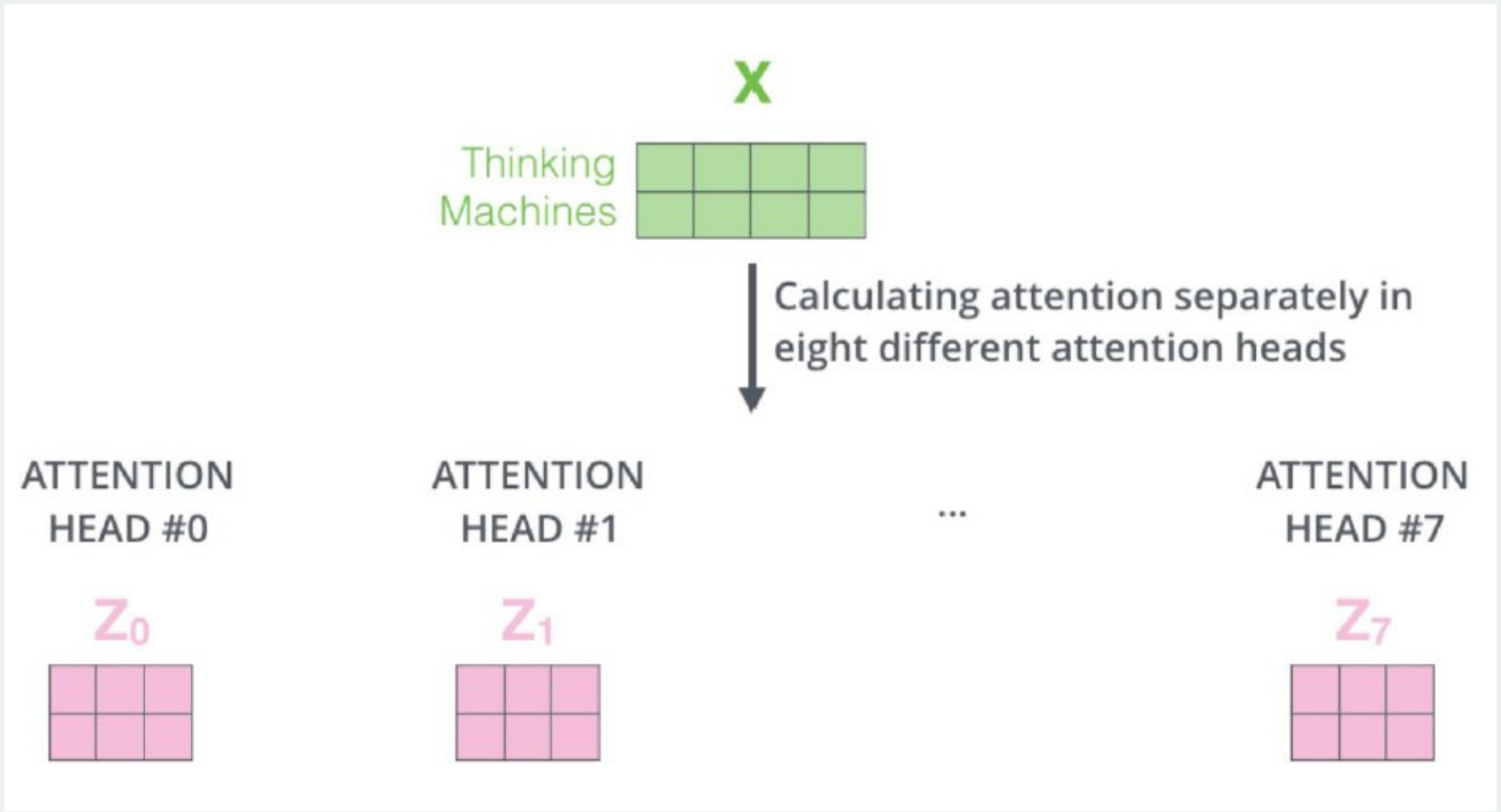
1 **Артур Кадурин, преподаватель**

Transformer



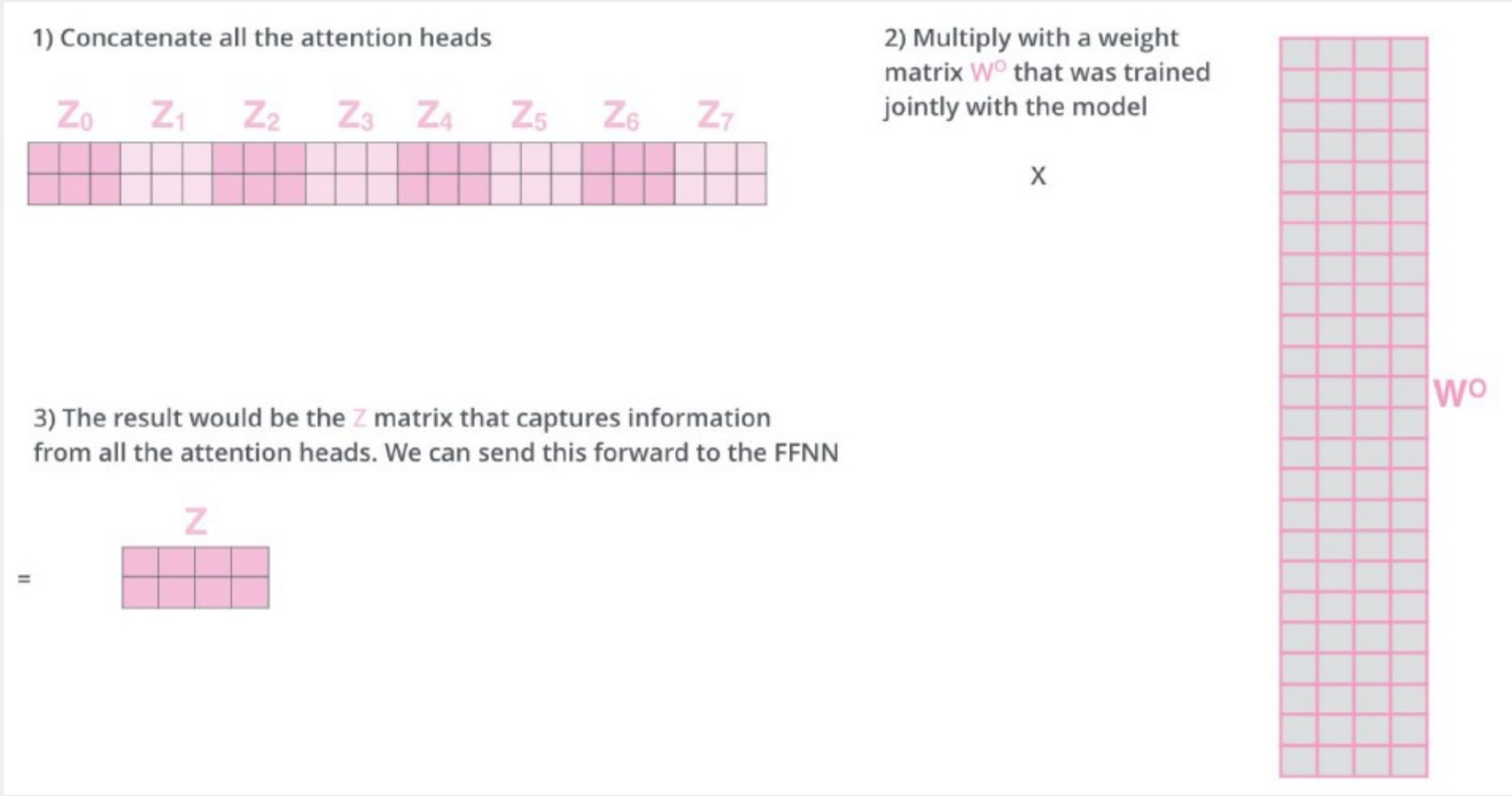
 Артур Кадурин, преподаватель

Transformer



1 **Артур Кадурин, преподаватель**

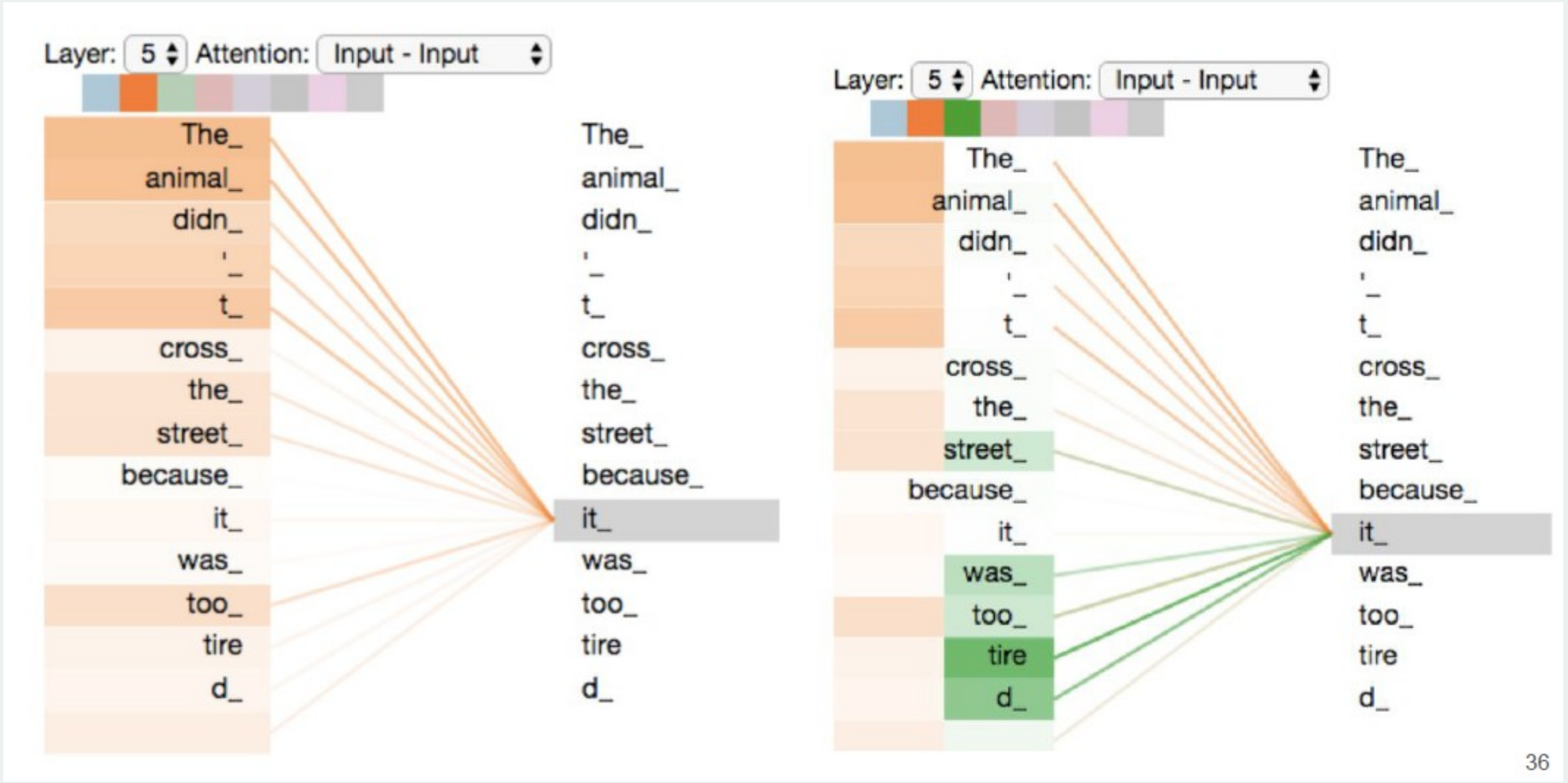
Transformer



1 **Артур Кадурин, преподаватель**

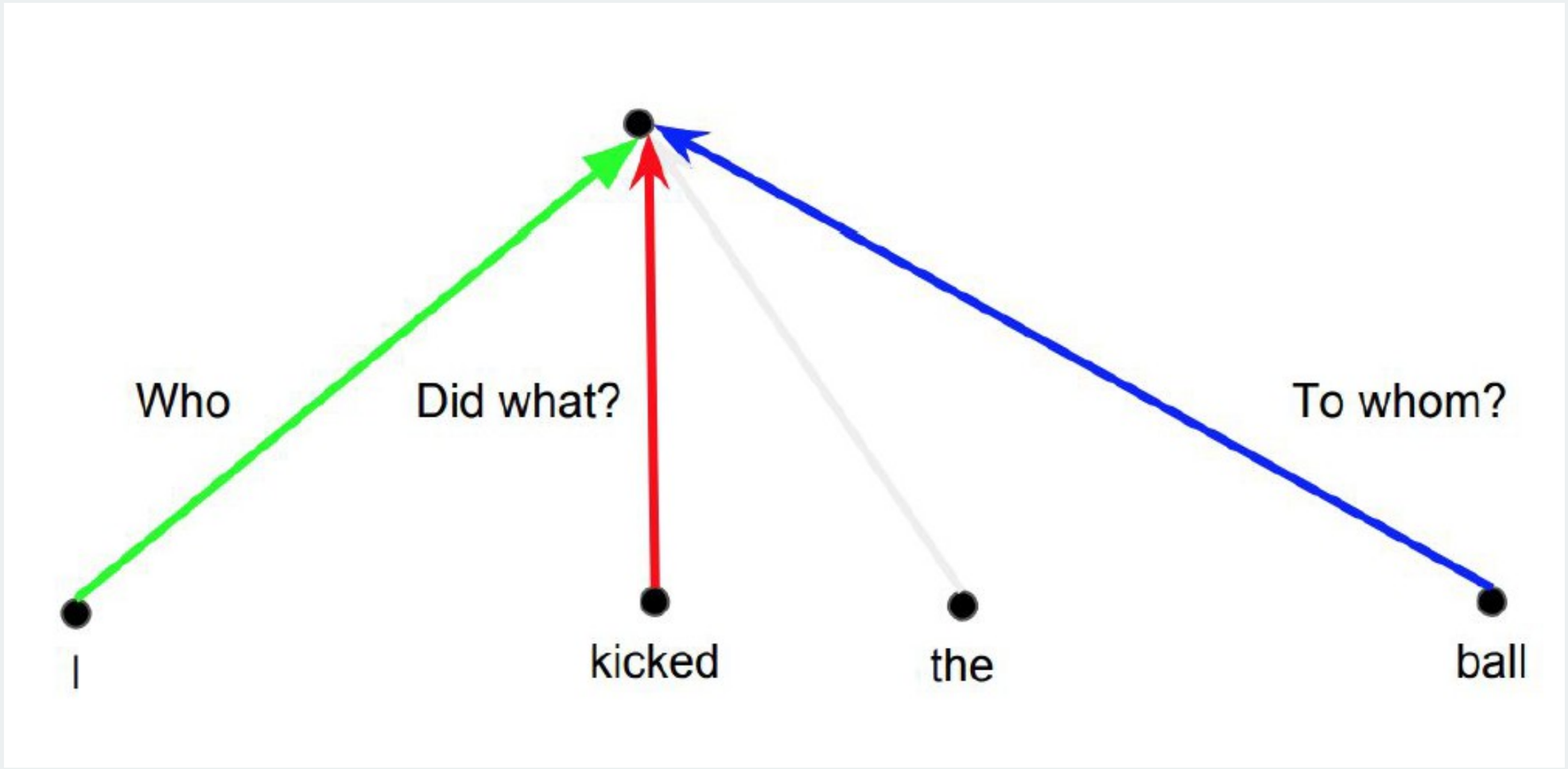
OTUS

Transformer



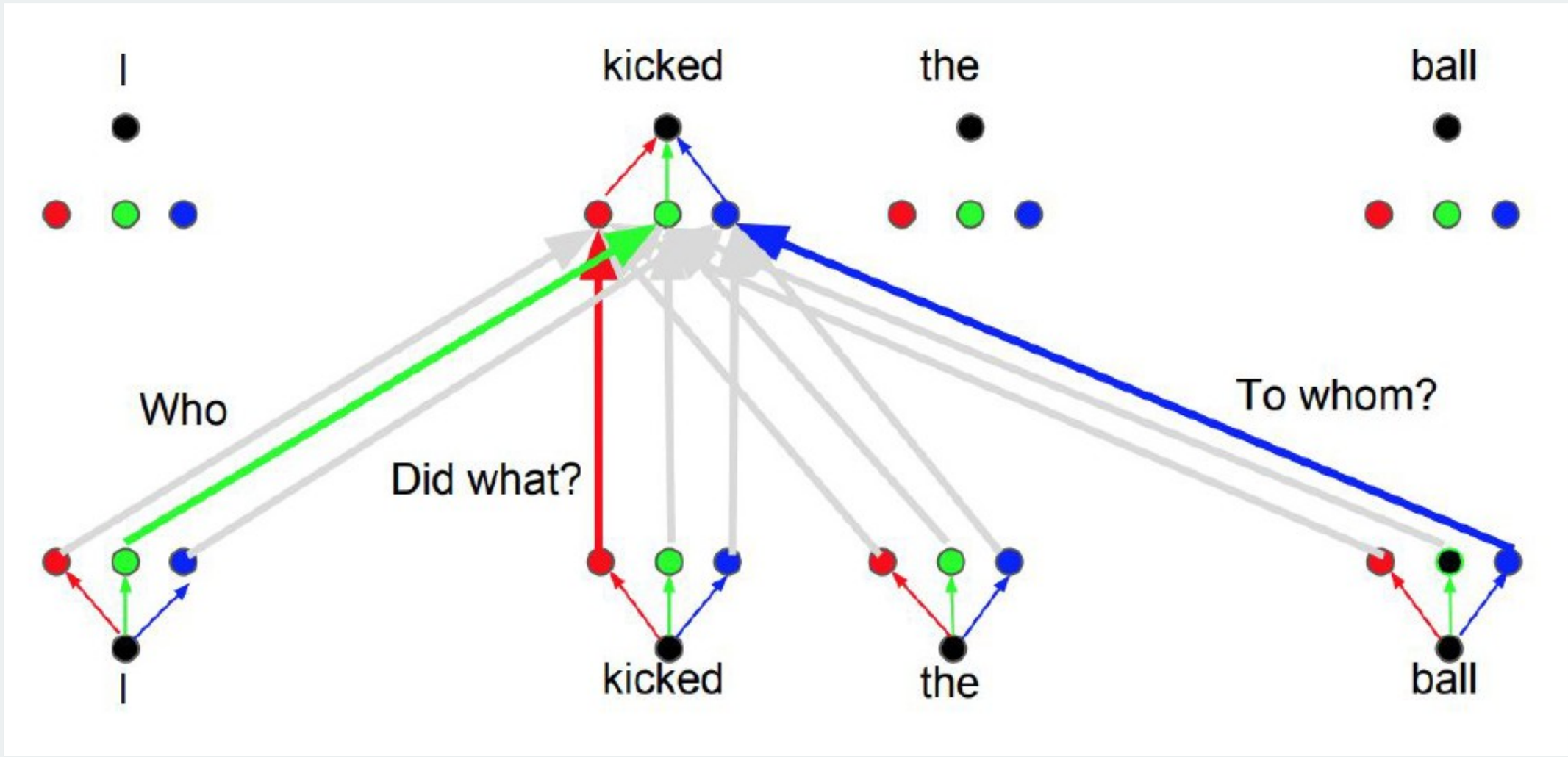
1 **Артур Кадурин, преподаватель**

Transformer



1 **Артур Кадурин, преподаватель**

Transformer



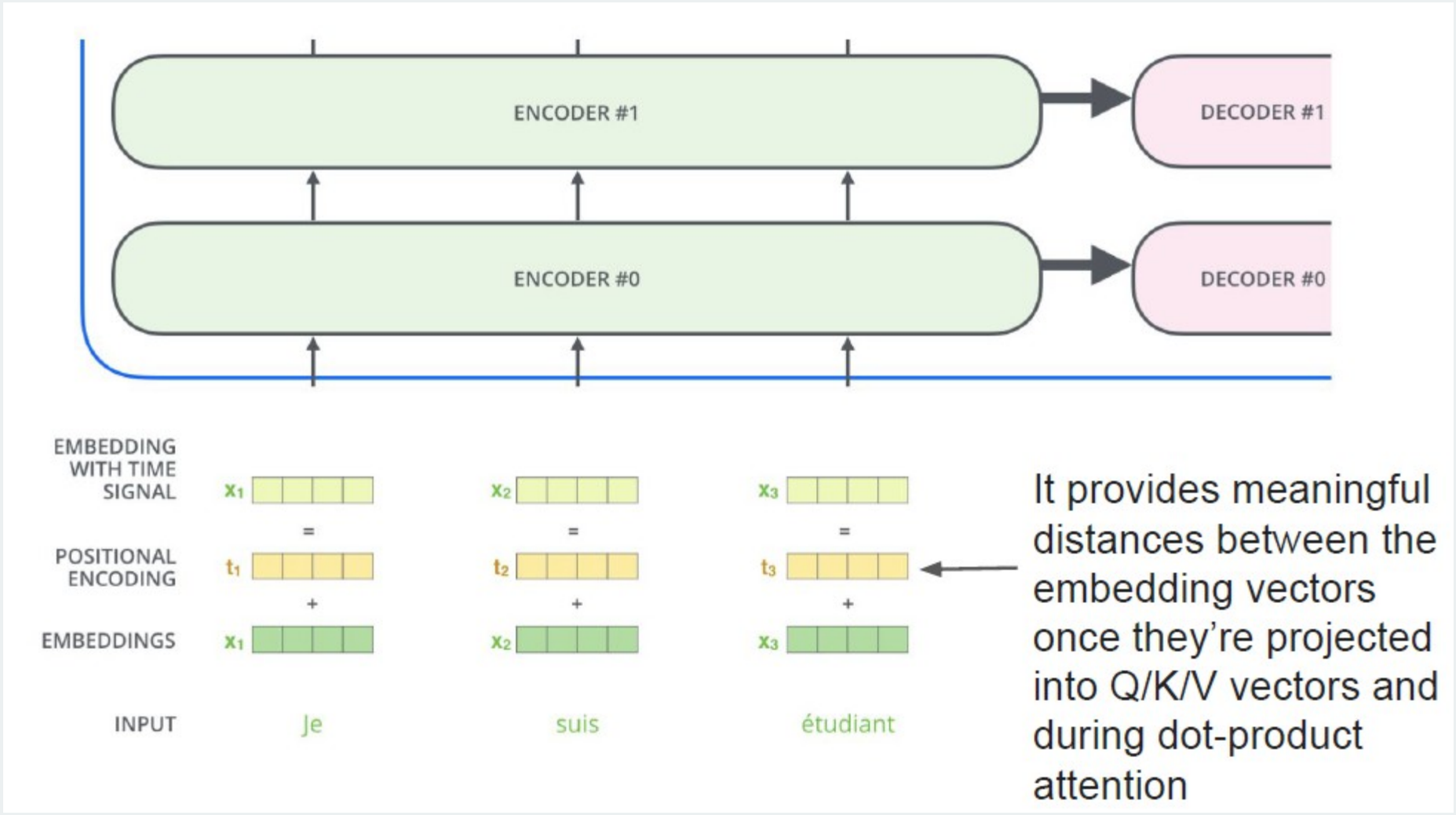
1 **Артур Кадурин, преподаватель**

Performance: WMT 2014 BLEU

	EN-DE	EN-FR
GNMT (orig)	24.6	39.9
ConvSeq2Seq	25.2	40.5
Transformer*	**28.4**	**41.8**

*Transformer models trained >3x faster than the others.

Transformer Positional Encoding



1 **Артур Кадурин, преподаватель**

Transformer - Positional Encoding

$$PE_{(pos,\ 2i)} = sin(pos\ /\ 10000^{2i\ /\ d_{\text{model}}})$$

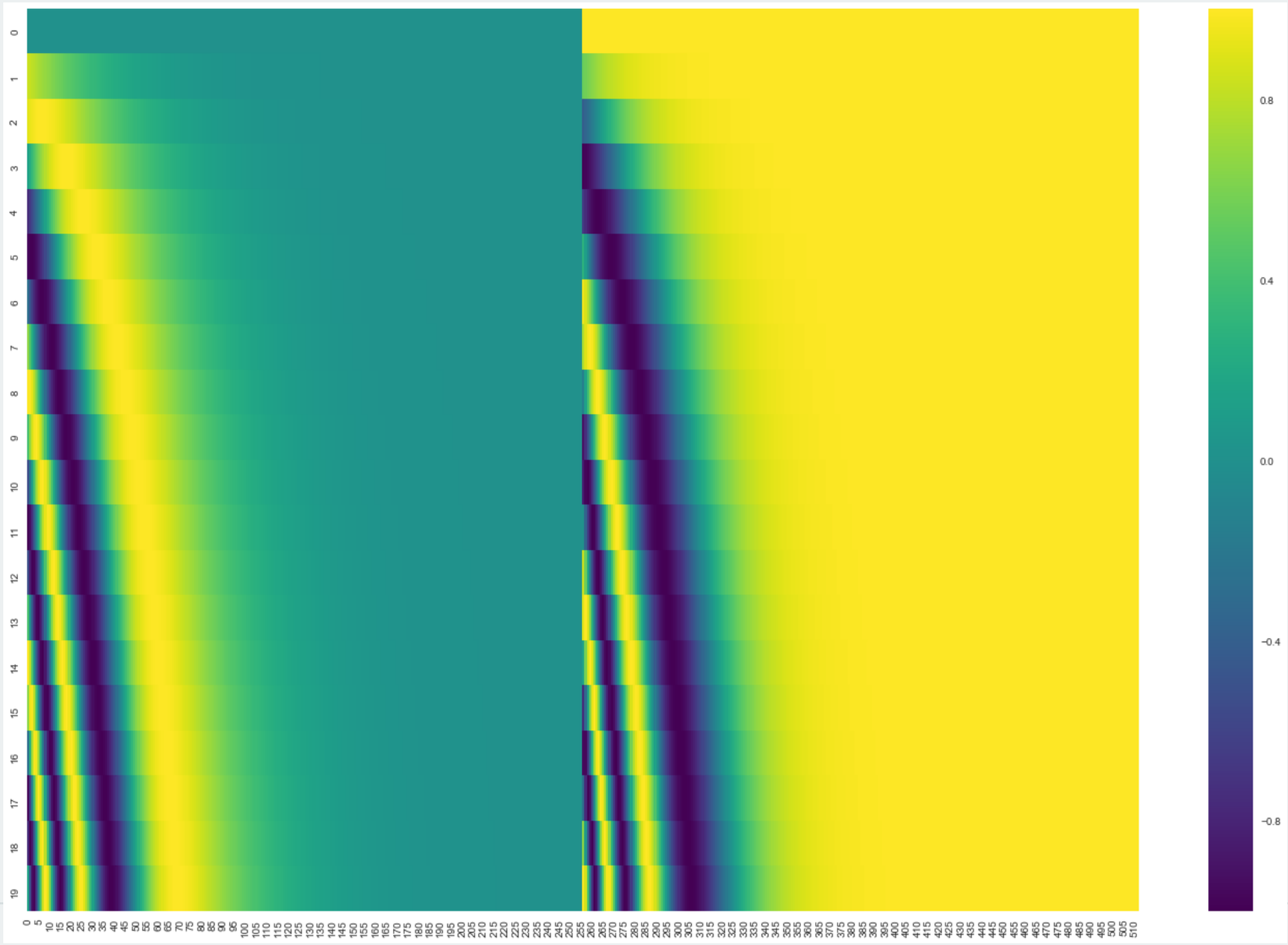
$$PE_{(pos,\ 2i+1)} = cos(pos\ /\ 10000^{2i\ /\ d_{\text{model}}})$$

- *pos* is the position
- *i* is the dimension.

Each dimension of the positional encoding corresponds to a sinusoid.
The wavelengths form a geometric progression from 2pi to 2pi * 10 000

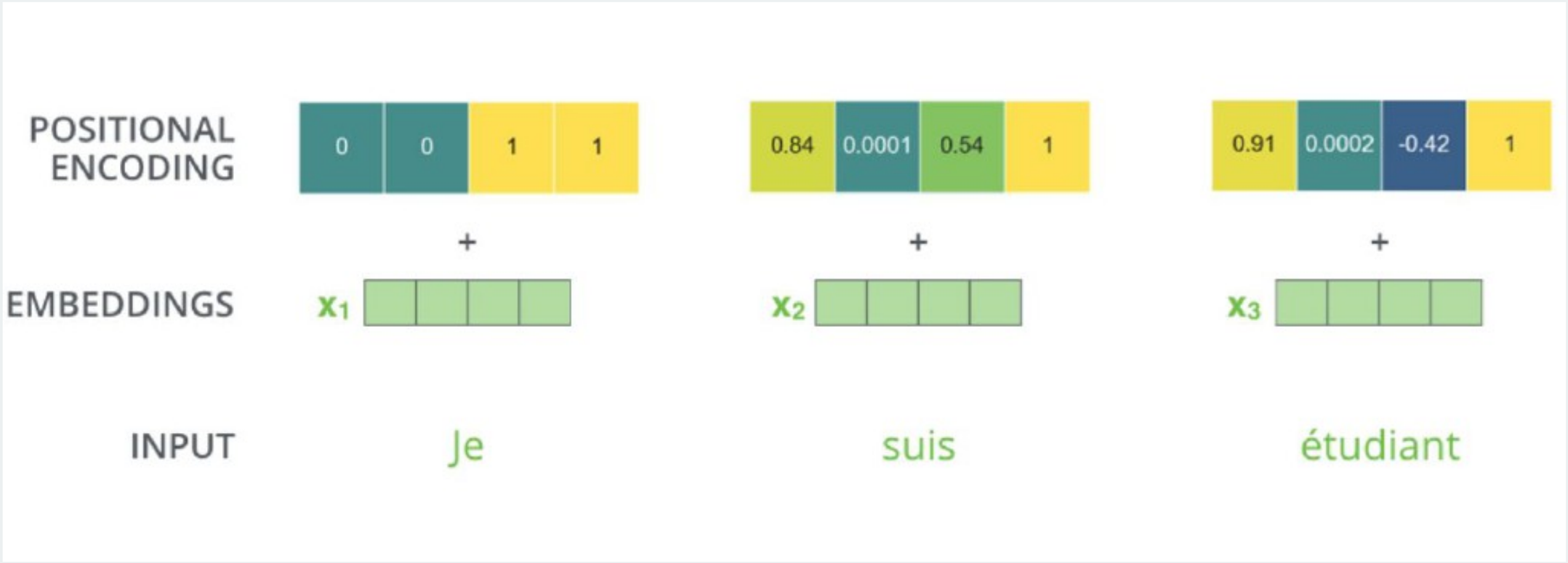
OTUS

Transformer - Positional Encoding



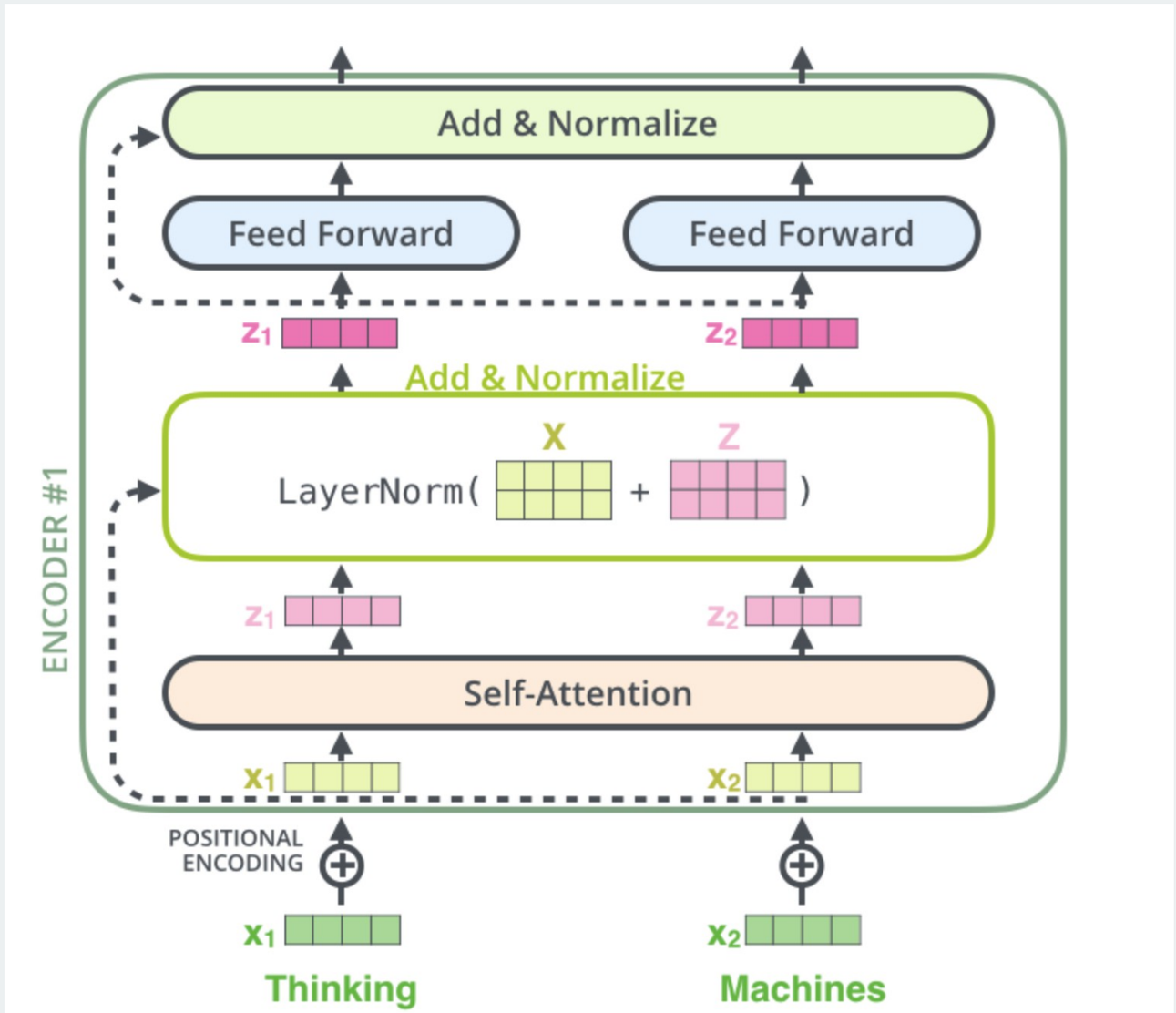
1 **Артур Кадурин, преподаватель**

Transformer - Positional Encoding



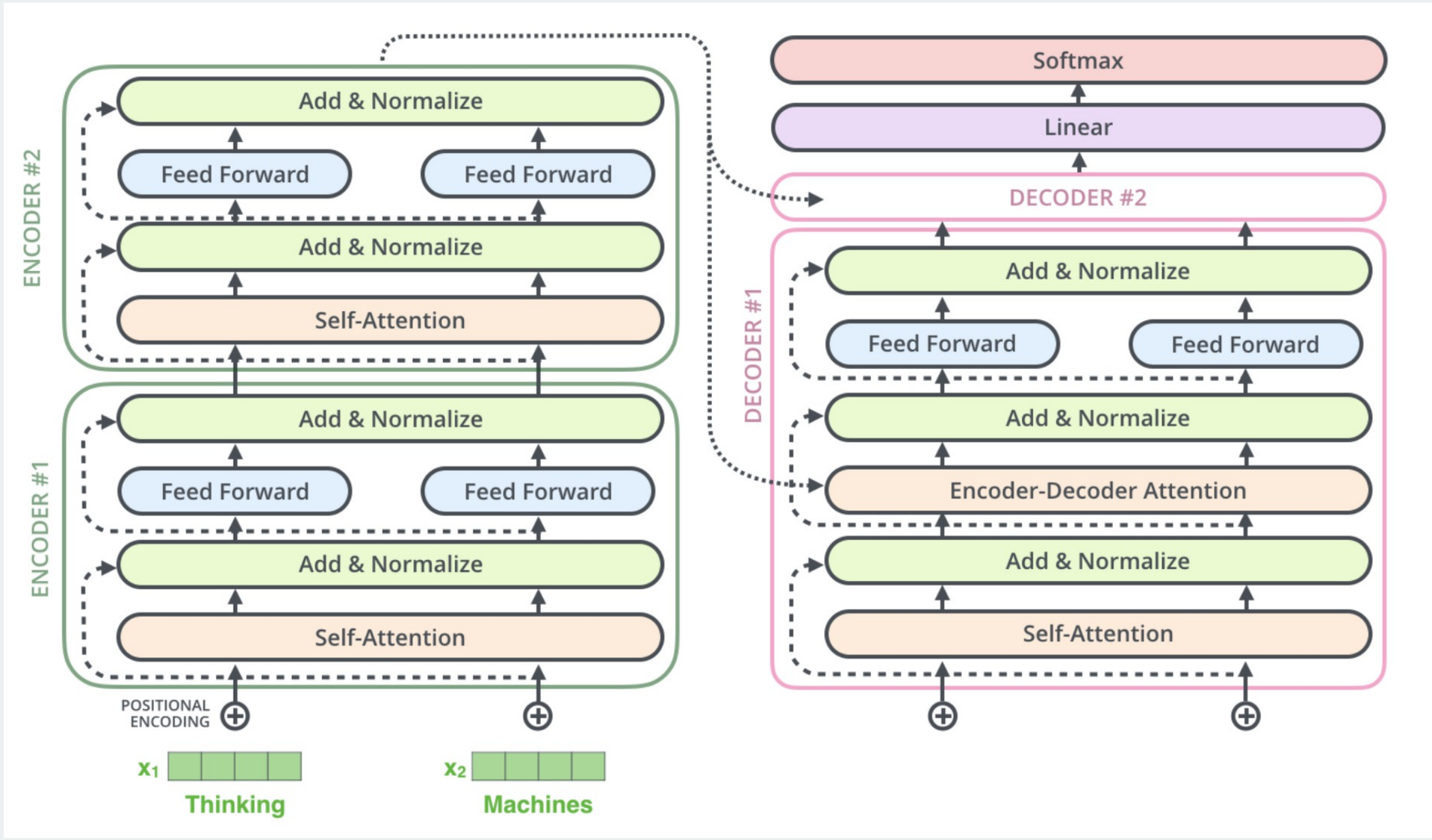
Артур Кадурин, преподаватель

Residuals & Add+Norm



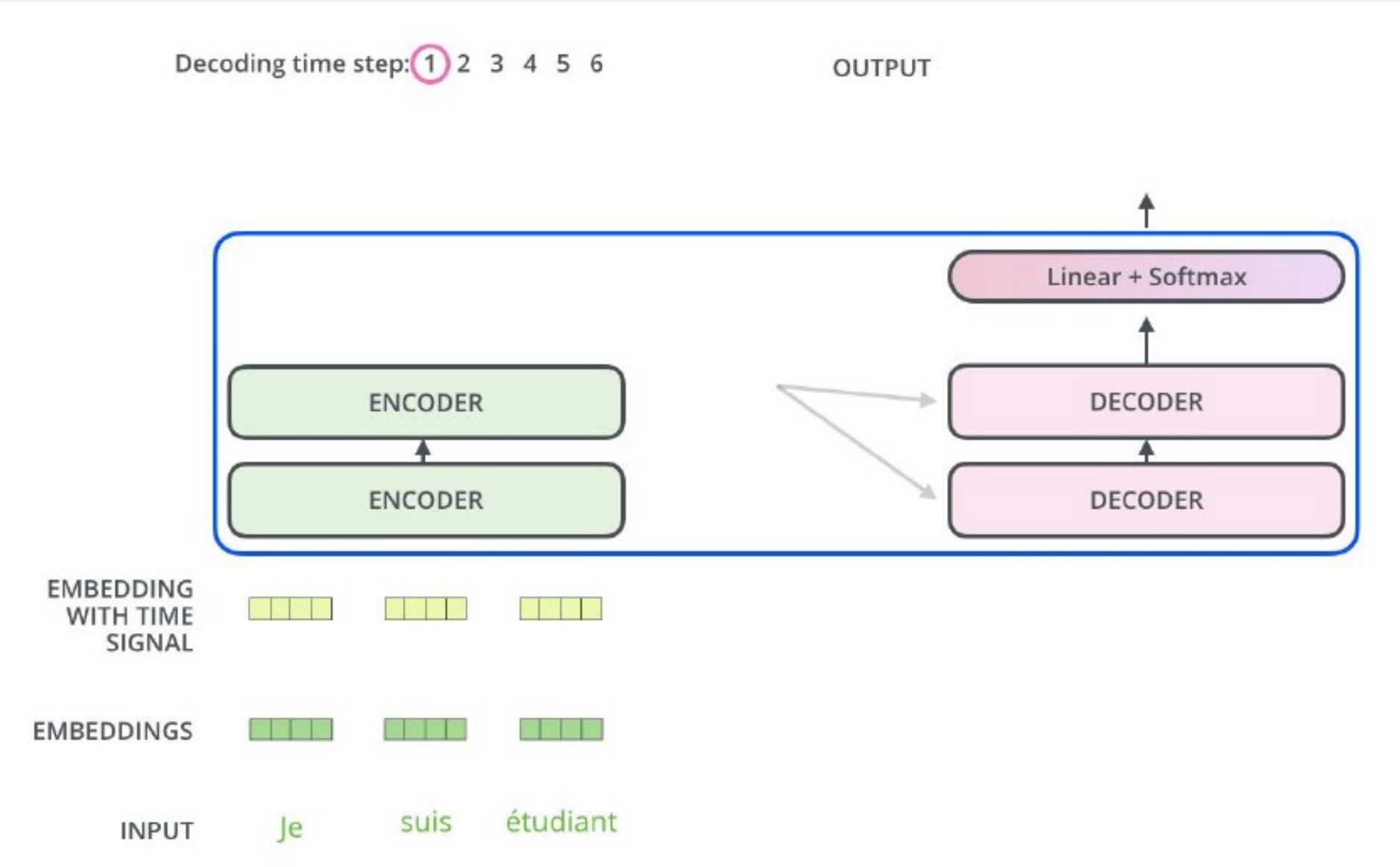
1 **Артур Кадурин, преподаватель**

Encoder & decoder



1 **Артур Кадурин, преподаватель**

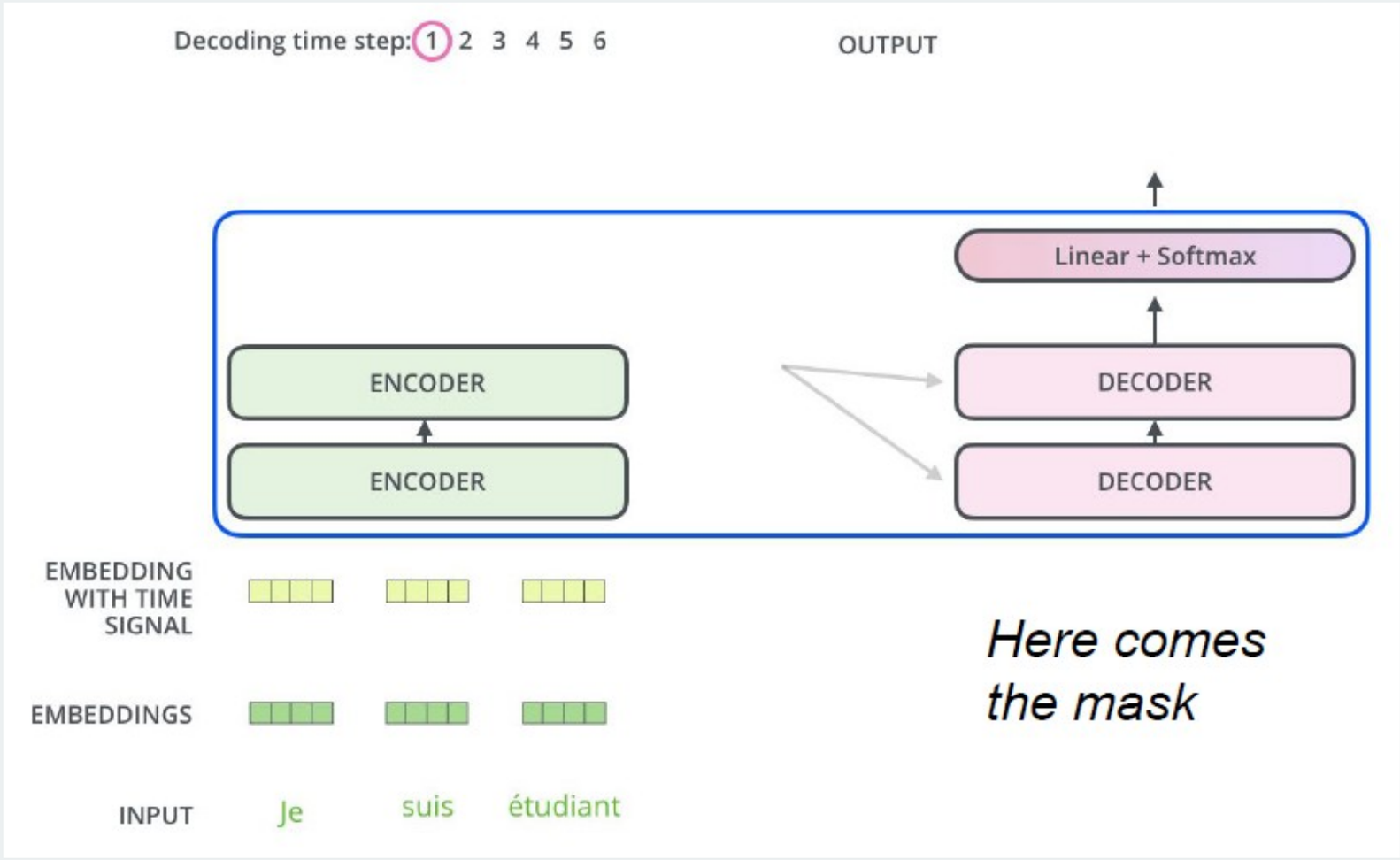
Transformer - Decoder



1 **Артур Кадурин, преподаватель**

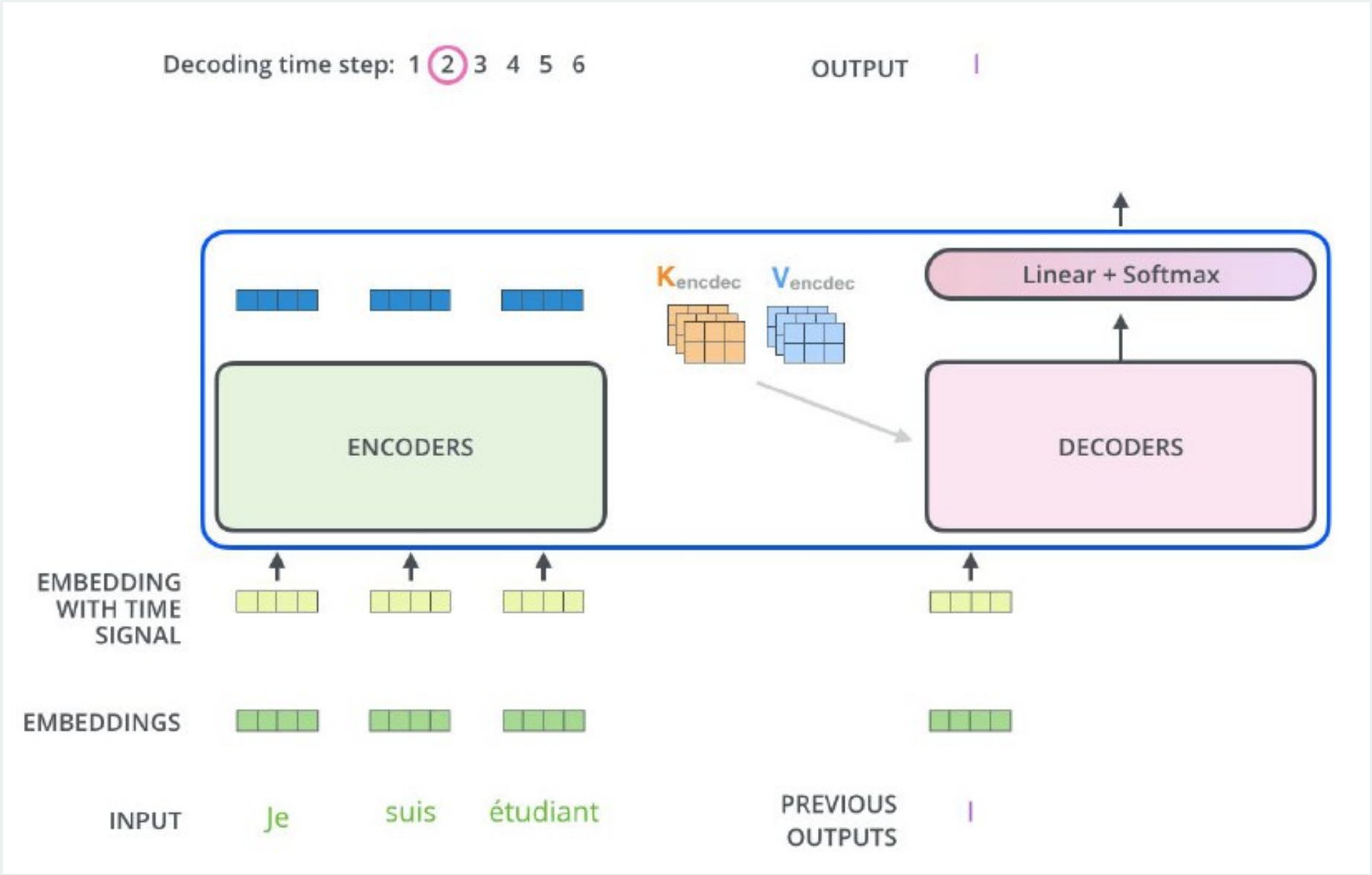
Decoder

masking future positions (setting them to -inf) before the softmax step in the self-attention calculation



Артур Кадурин, преподаватель

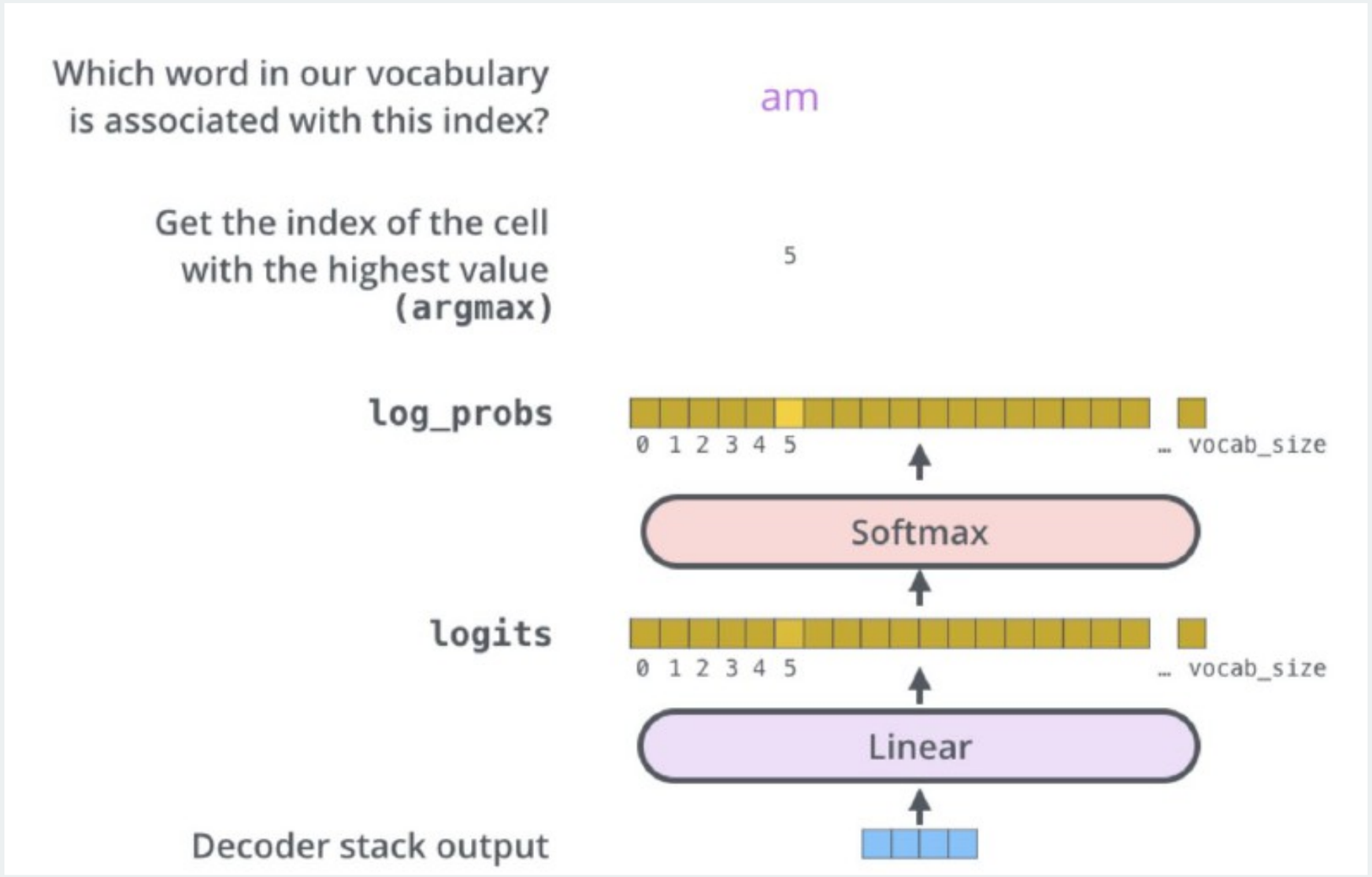
Transformer-decoder



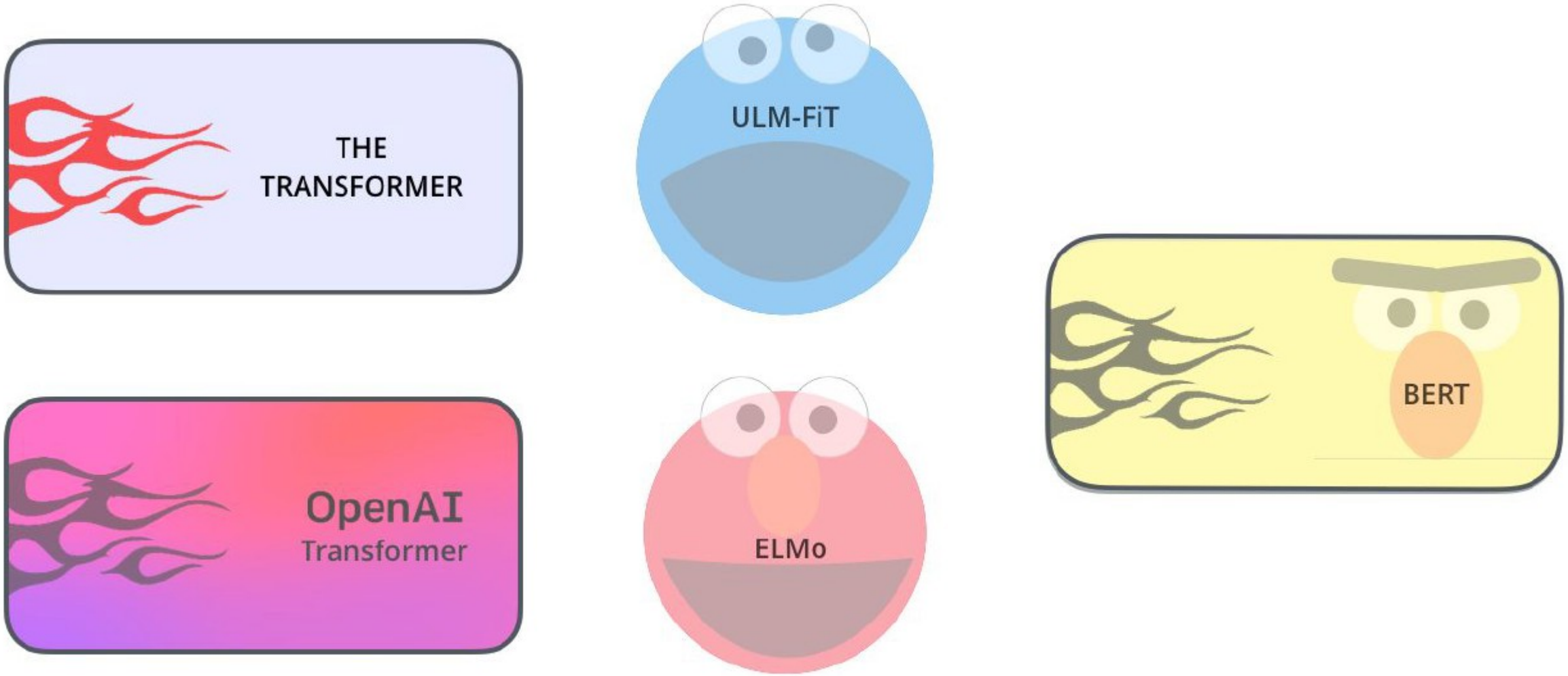
1 **Артур Кадурин, преподаватель**

OTUS

Transformer - Final Linear and Softmax Laye



1 **Артур Кадурин, преподаватель**

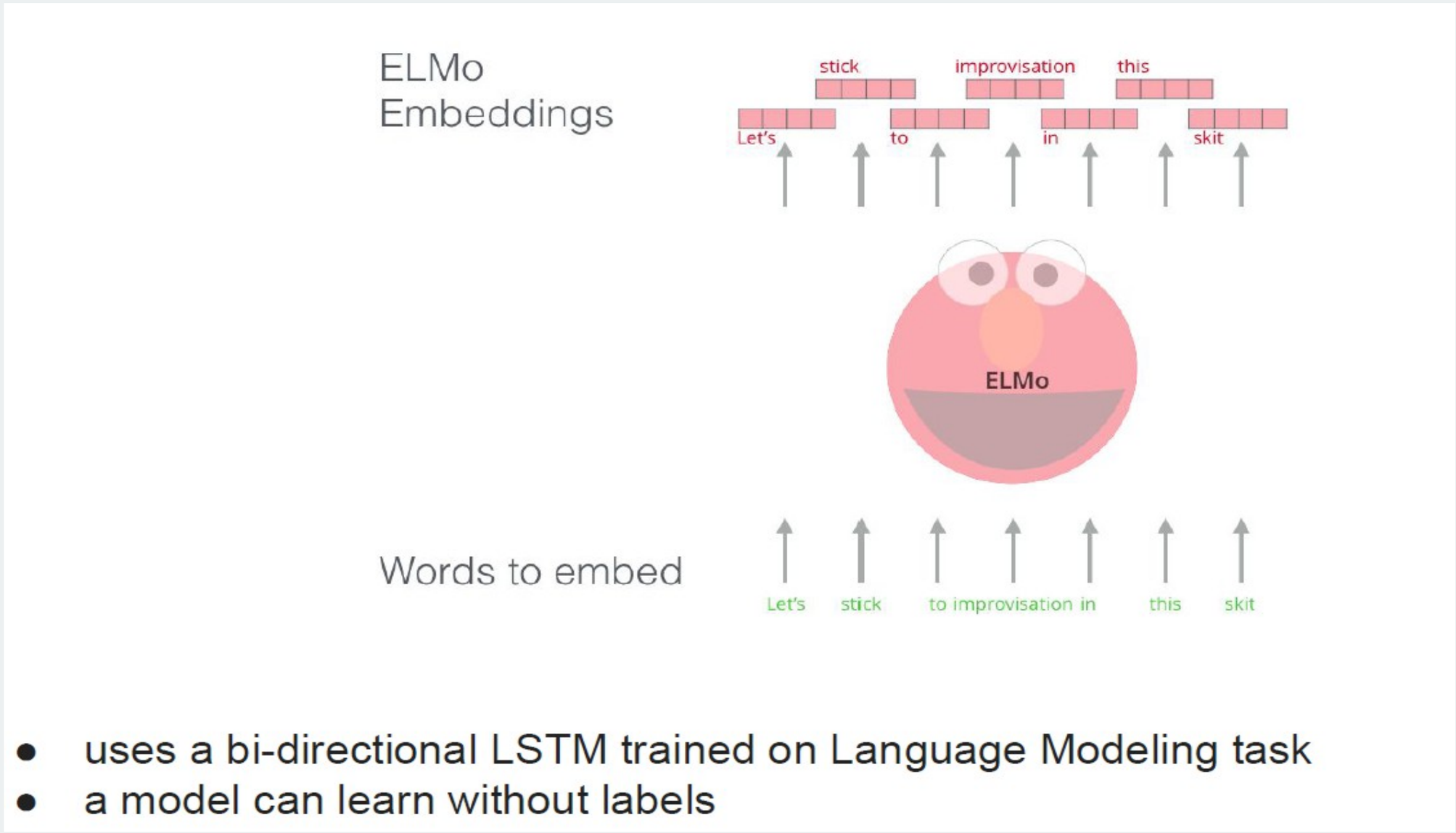


1 Артур Кадурин, преподаватель

ELMo
Embeddings from Language Models

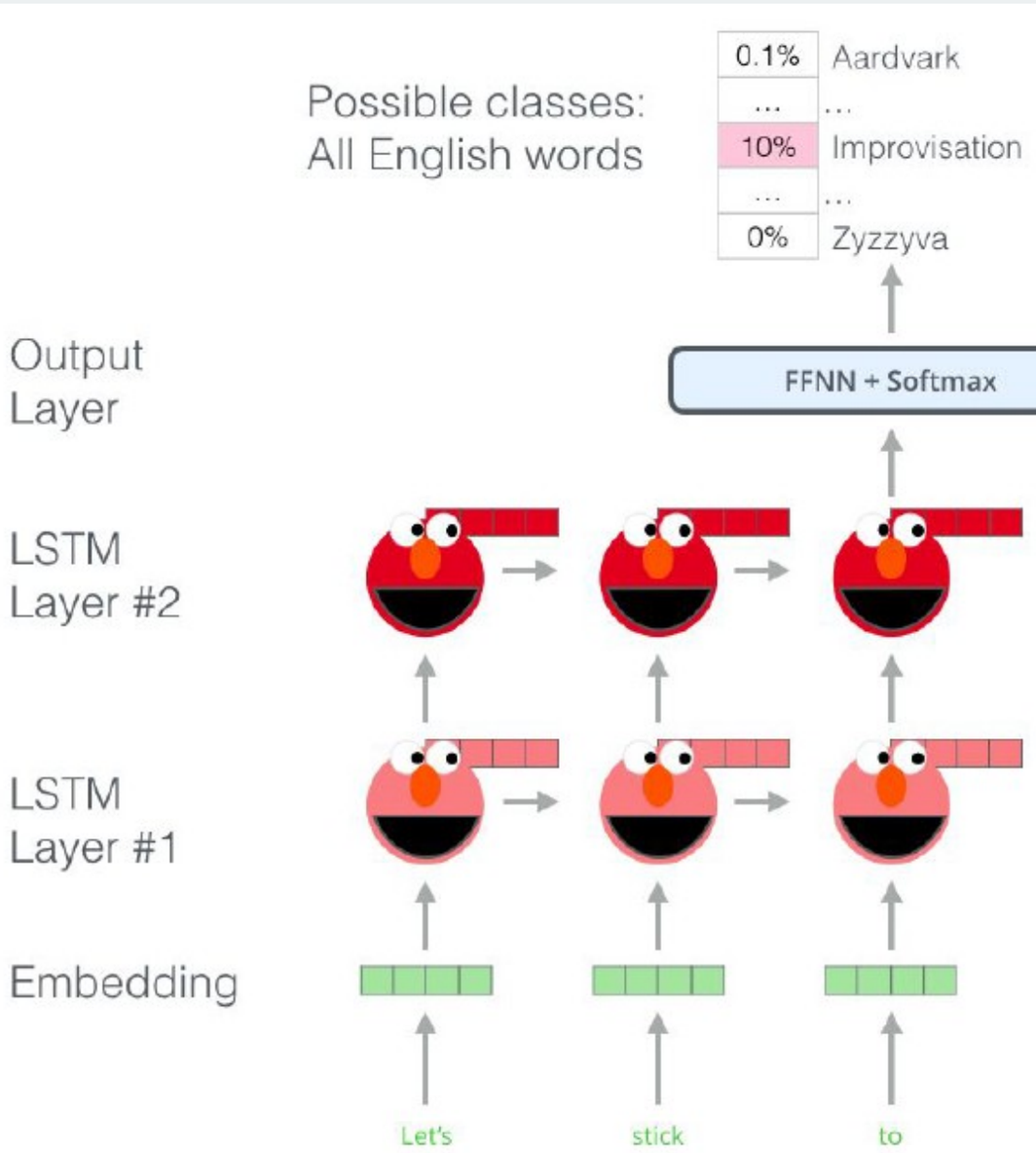
1 **Артур Кадурин, преподаватель**

ELMo: Contextualized word embeddings



Артур Кадурин, преподаватель

Bidirectional Language Models (biLMs)



biLMs consist of forward and backward LMs:

- forward:

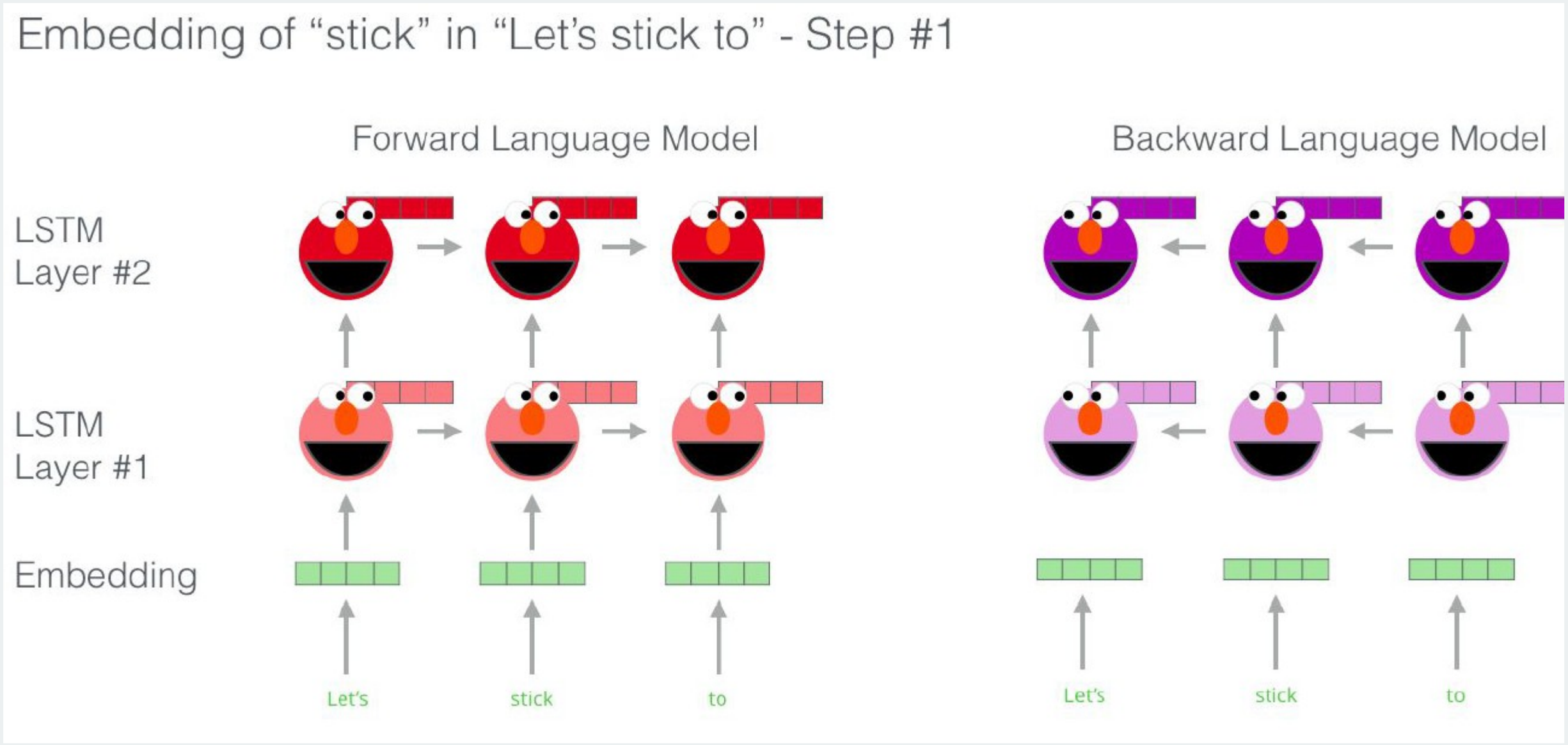
$$p(t_1,t_2,\ldots,t_N)=\prod_{k=1}^{N} p(t_k|t_1,t_2,\ldots,t_{k-1})$$

- Backward:

$$p(t_1,t_2,\ldots,t_N)=\prod_{k=1}^{N} p(t_k|t_{k+1},t_{k+2},\ldots,t_N)$$

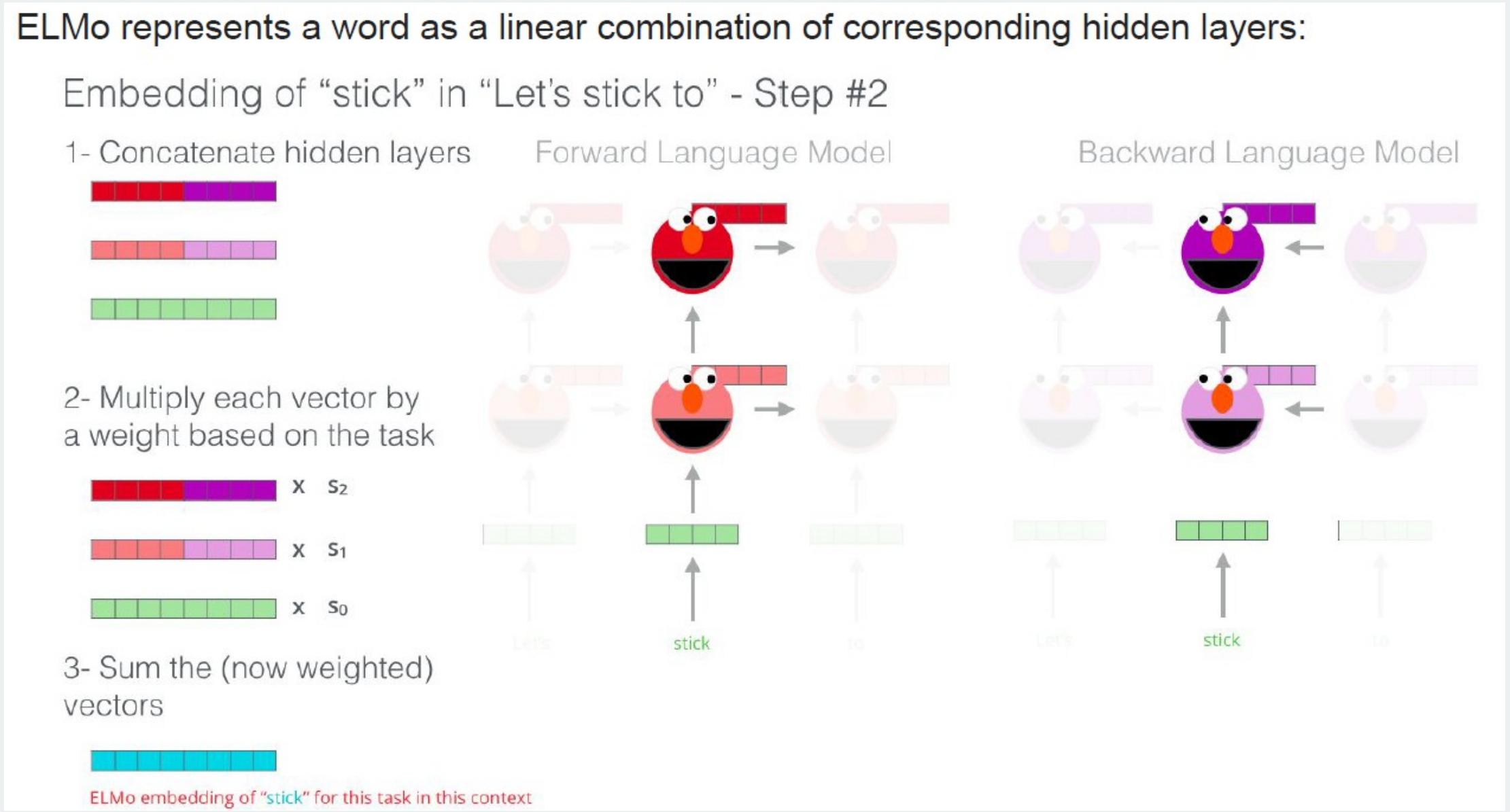
LSTM predicts next word in both directions to build biLMs

ELMo: main pipeline



 Артур Кадурин, преподаватель

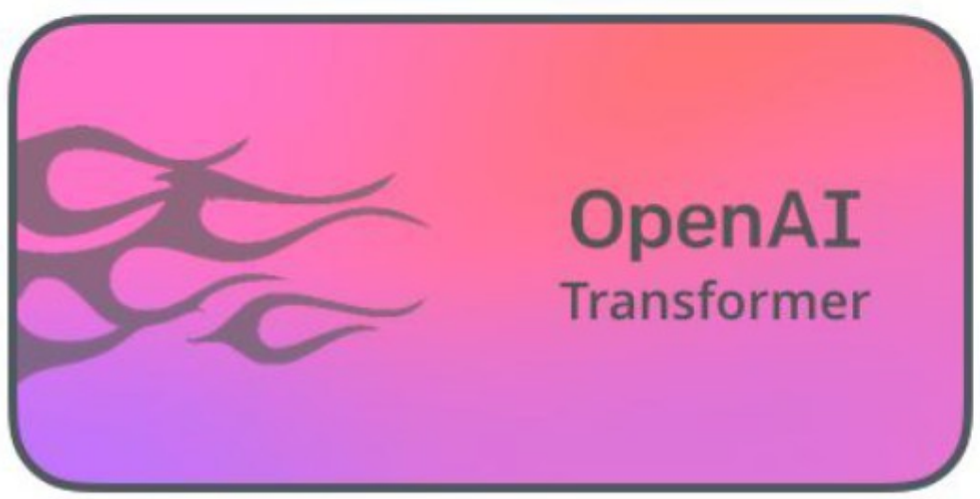
ELMo: main pipeline



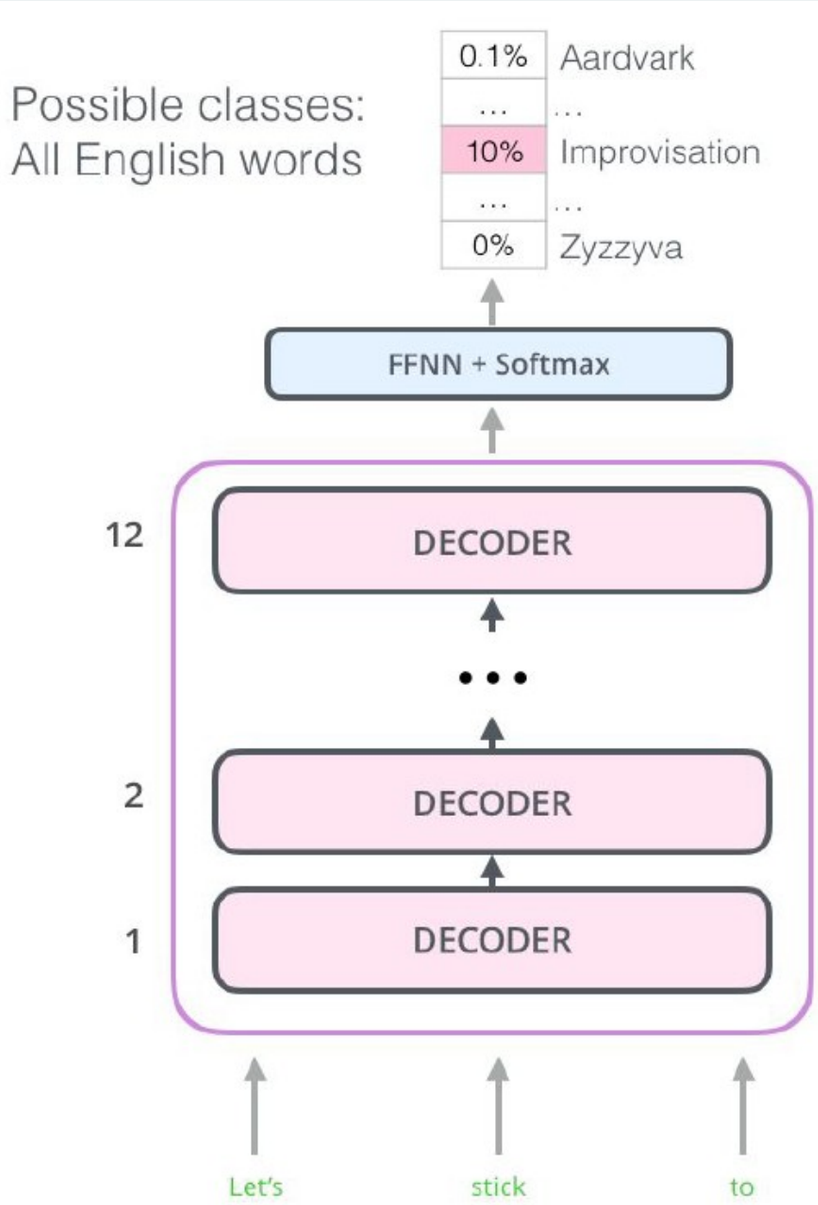
Артур Кадурин, преподаватель

OpenAI Transformer

- The Encoder-Decoder structure of the transformer made it perfect for machine translation
- But what about sentence classification?
- **Main goal: pre-train a language model that can be fine-tuned for other tasks**



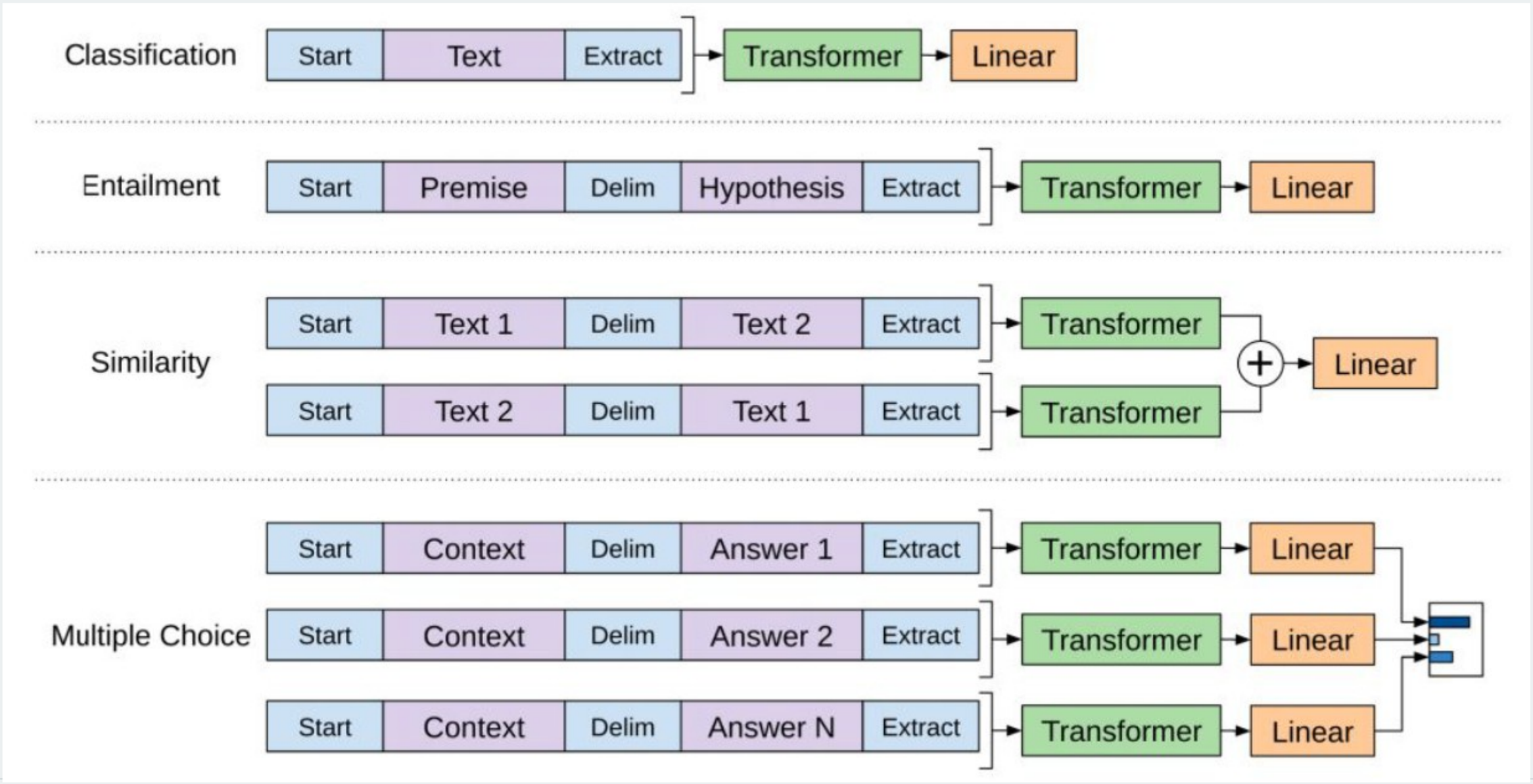
OpenAI Transformer



Differences from vanilla Transformer:

- no encoder
- decoder layers would not have the encoder-decoder attention sublayer
- Pre-train the model on predicting the next word using massive (unlabeled) datasets

nput transformations for different tasks



 Артур Кадурин, преподаватель

OpenAI Transformer

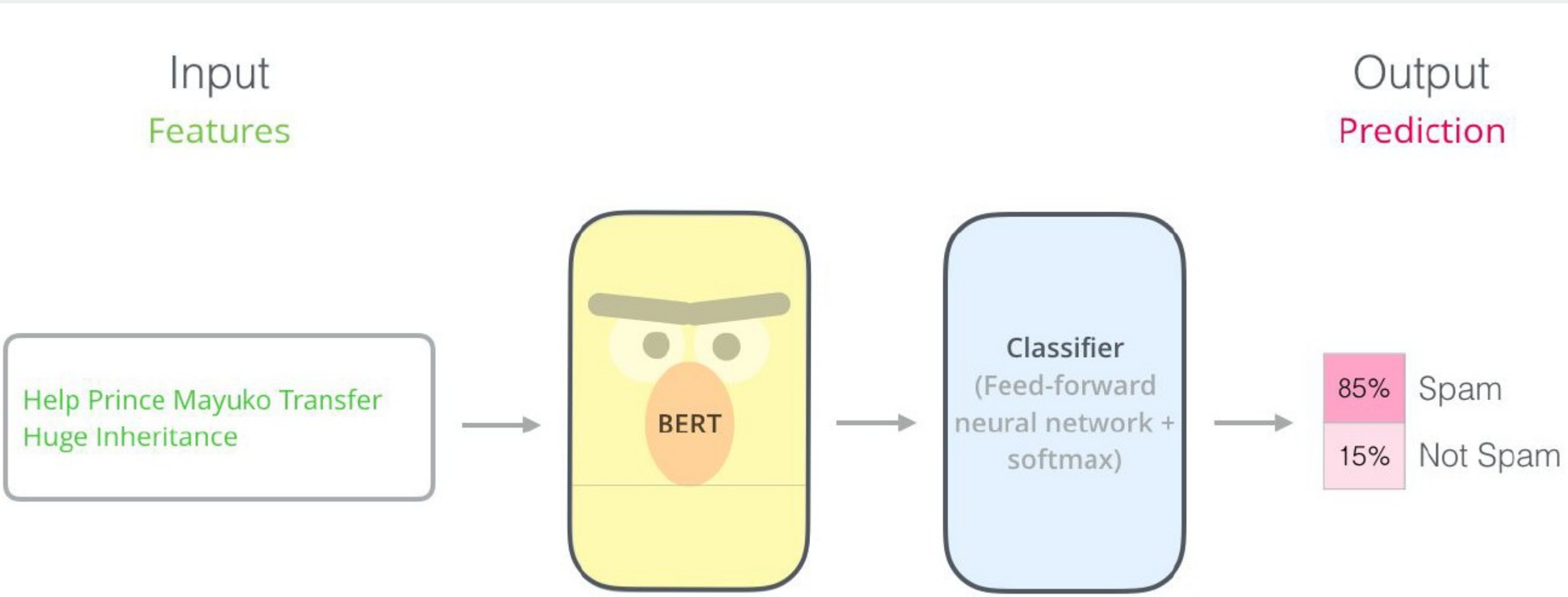
- Transformer? Yes
- Bi-directional? No :(

OpenAI Transformer

- Mask 15% of the input
- Predict only masked tokens
- *Cloze* task (Taylor, 1953)

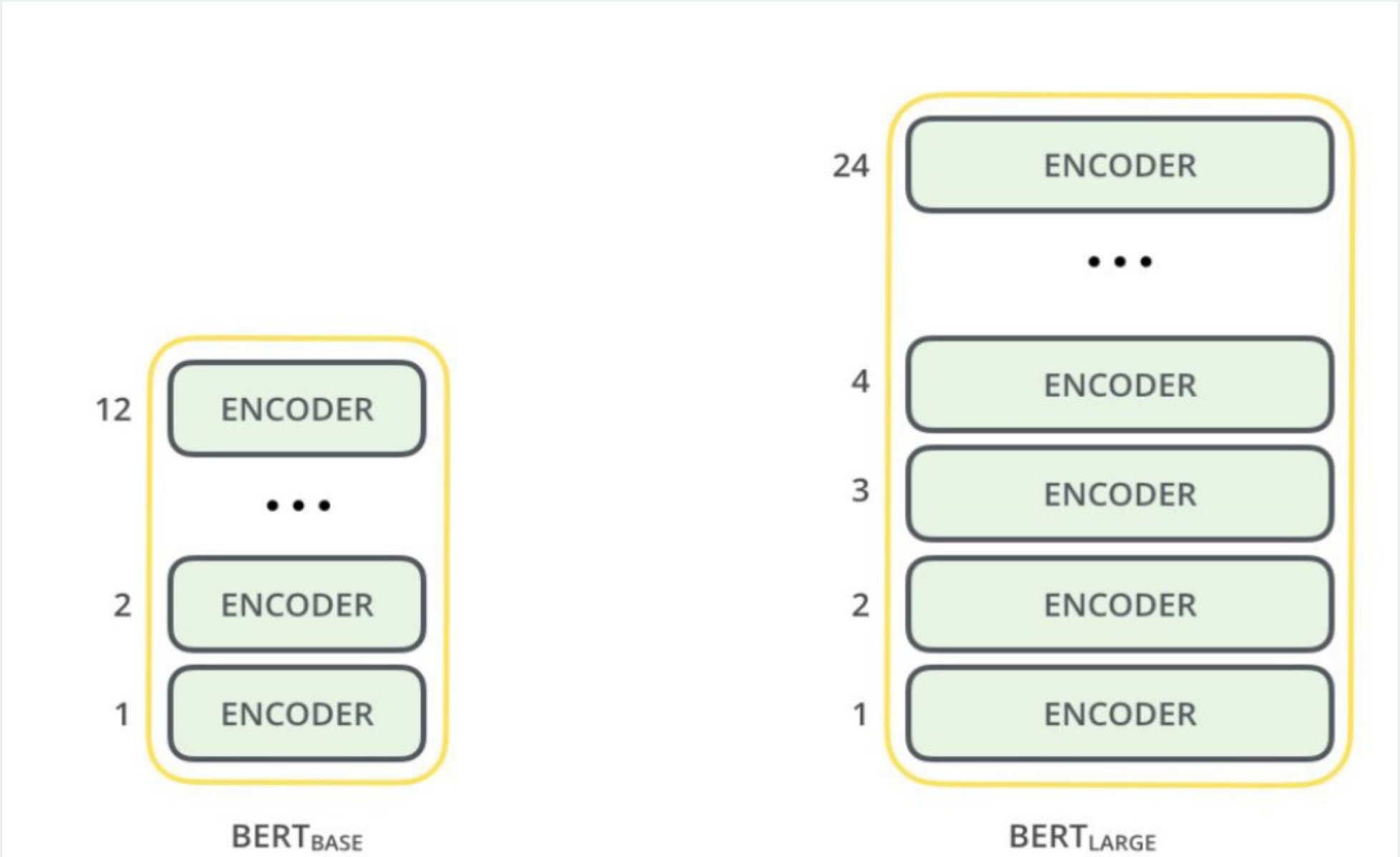
OTUS

BERT
Bidirectional Encoder Representations from Transformers



1 **Артур Кадурин, преподаватель**

BERT

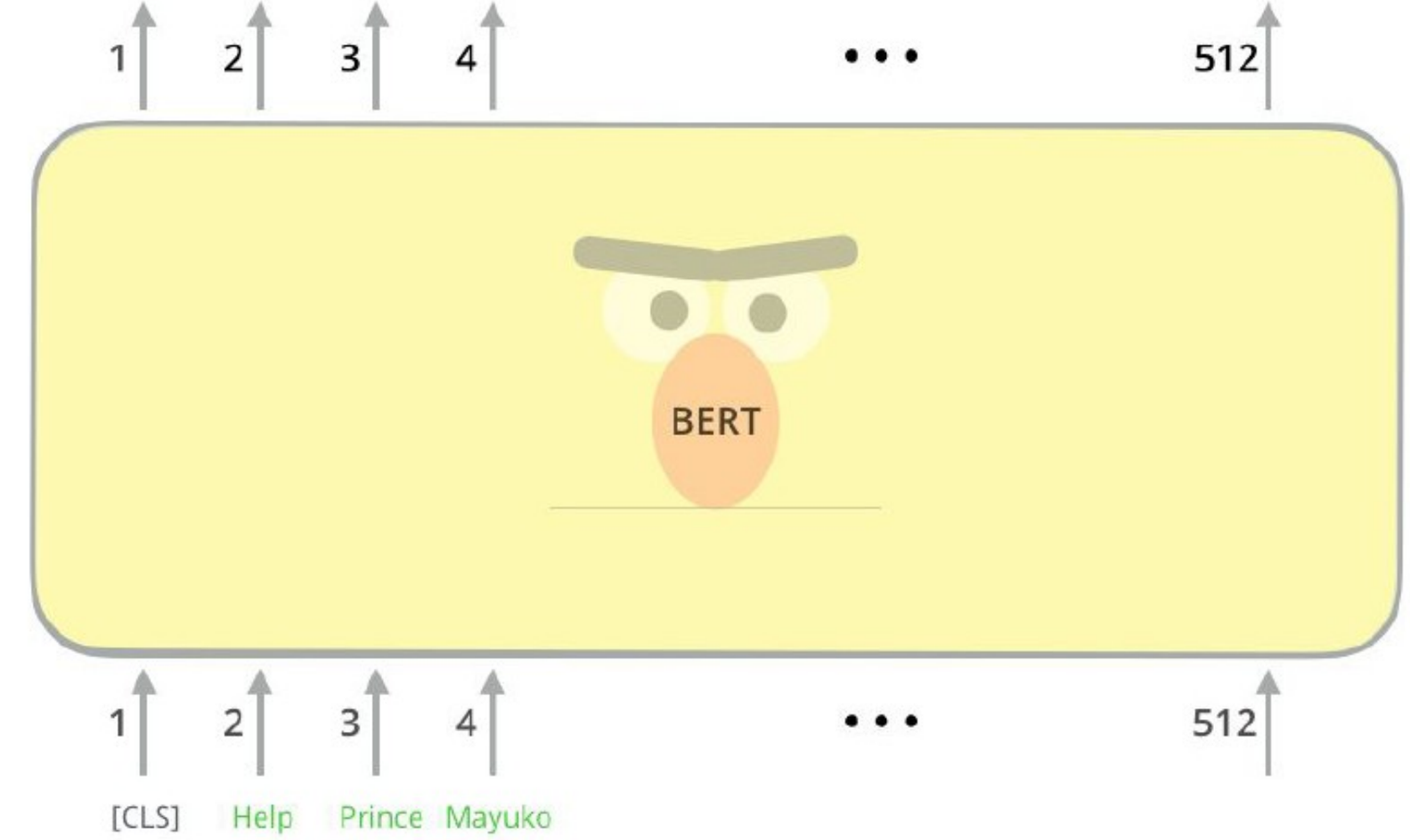


1 **Артур Кадурин, преподаватель**

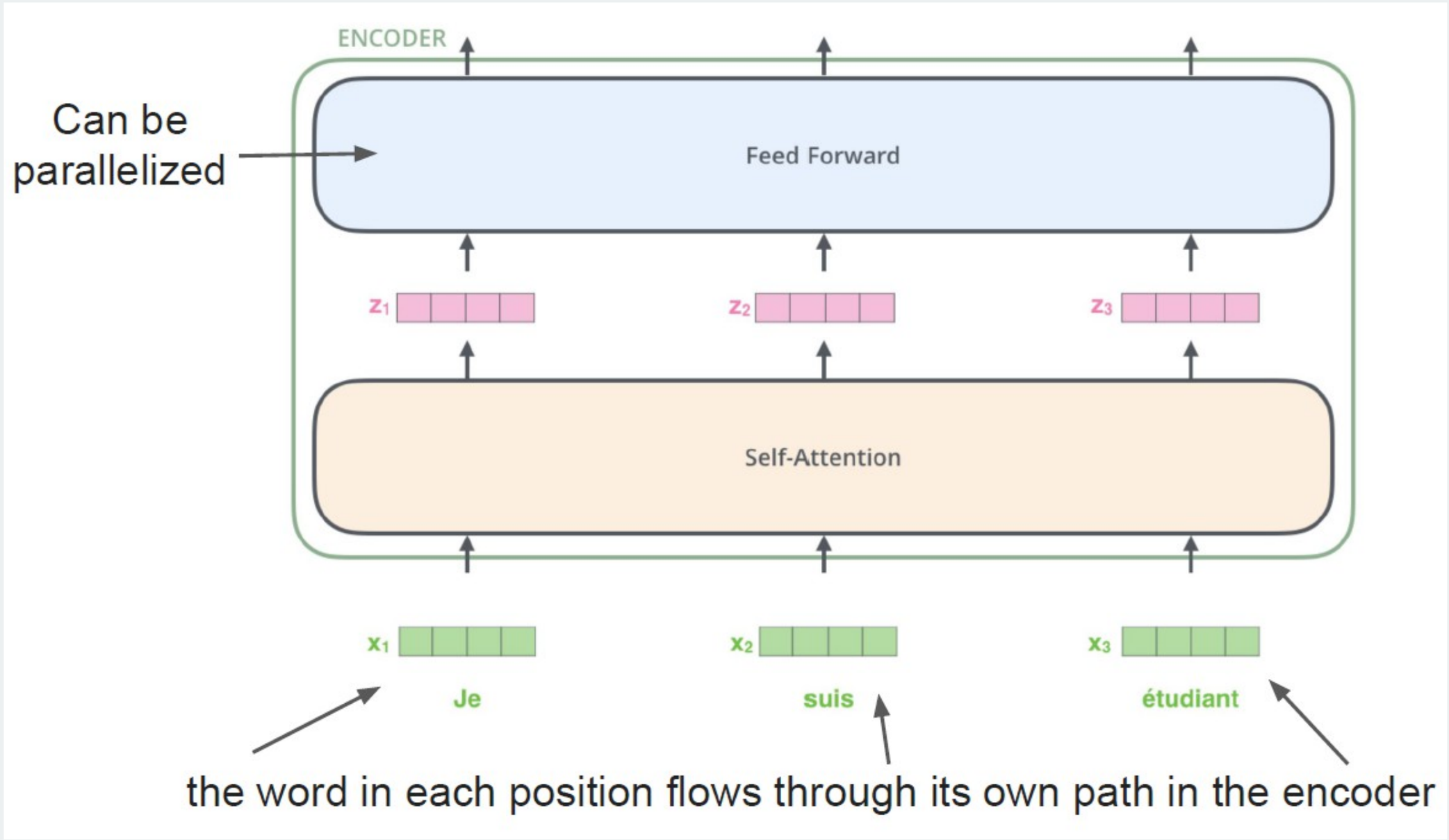
BERT vs. Transformer

			
		Base BERT	**Large BERT**
Encoders	6	12	24
Units in FFN	512	768	1024
Attention Heads	8	12	16

BERT - Model inputs

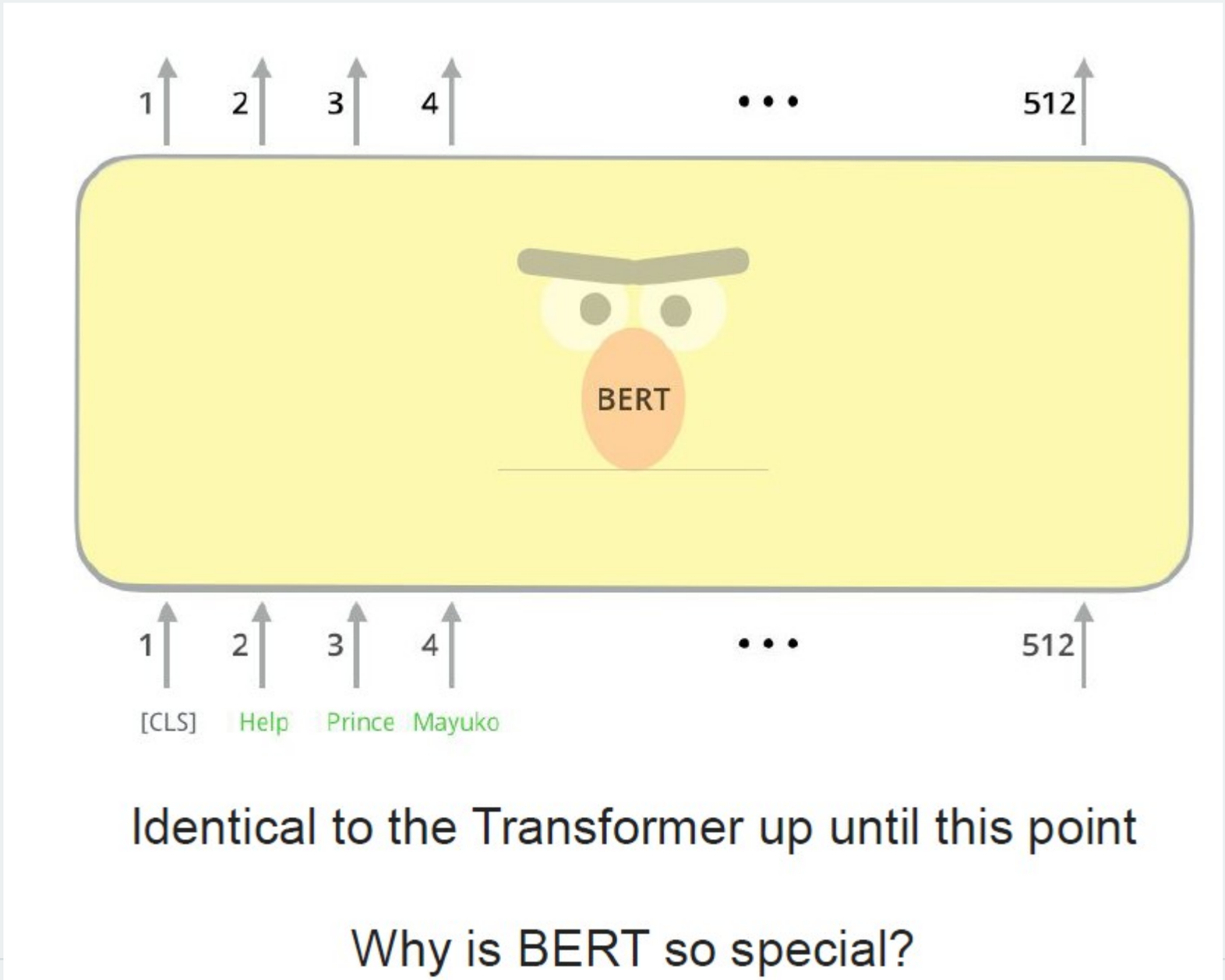


Transformer Block in BERT

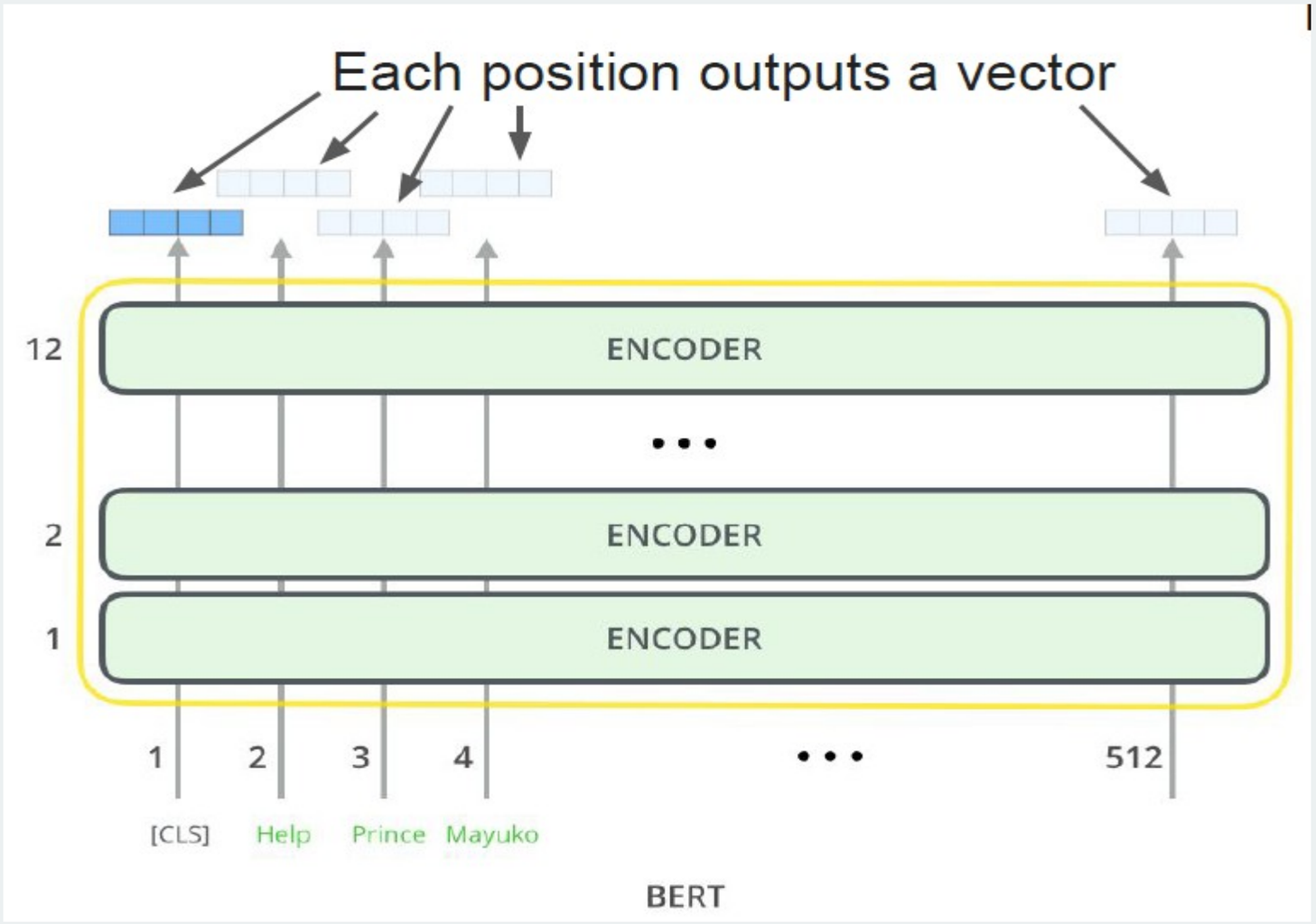


1 **Артур Кадурин, преподаватель**

BERT - Model inputs



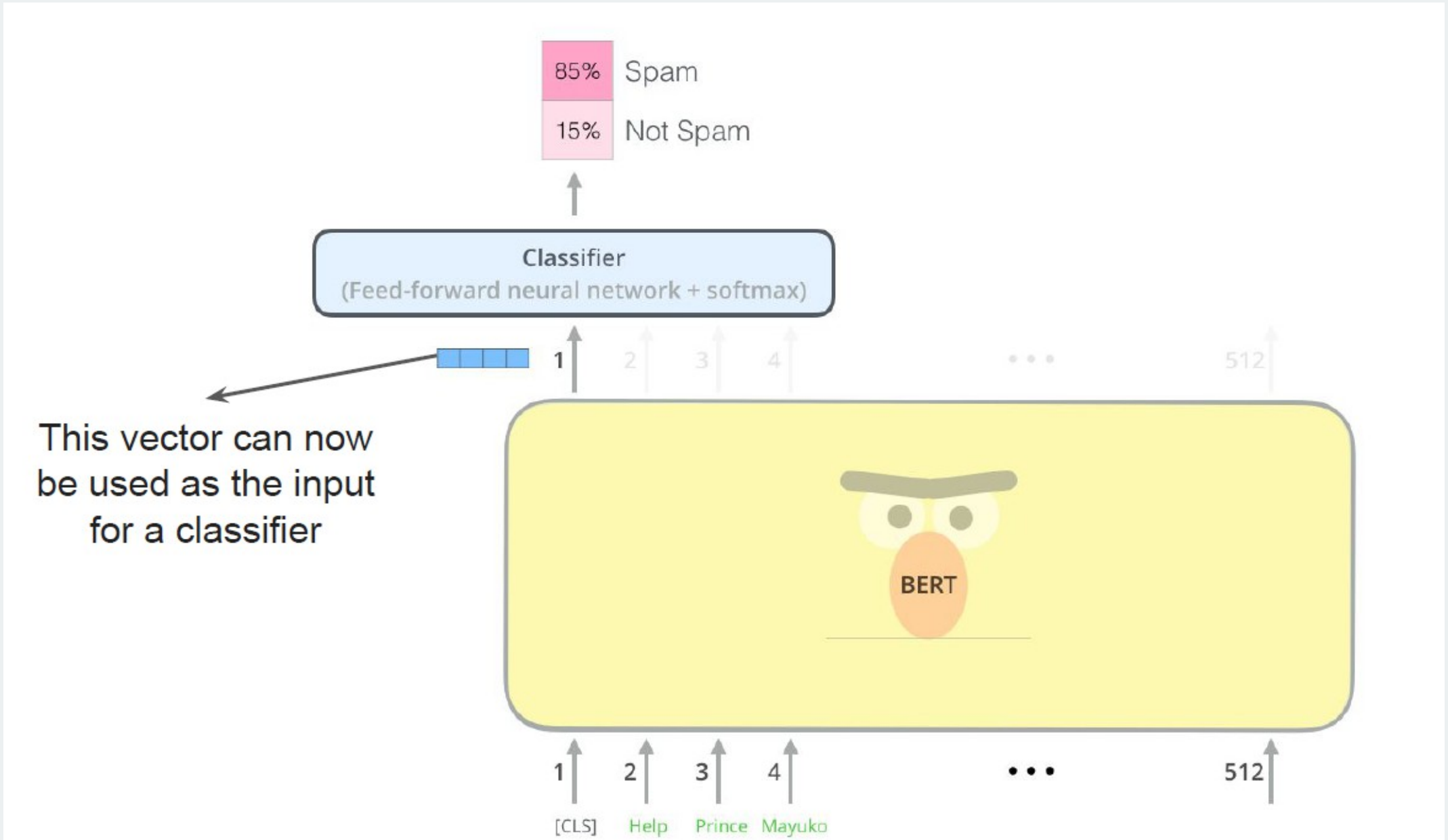
BERT



For sentence classification we focus on the first position (that we passed [CLS] token to)

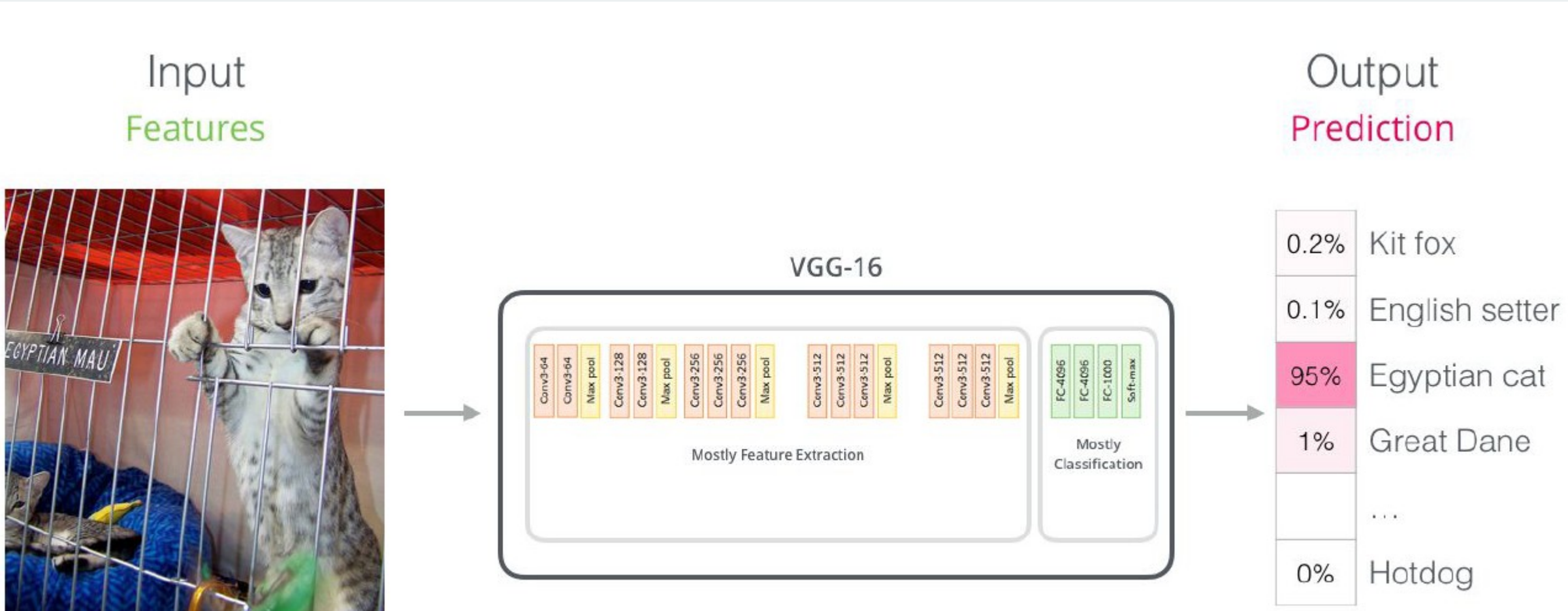
Артур Кадурин, преподаватель

BERT

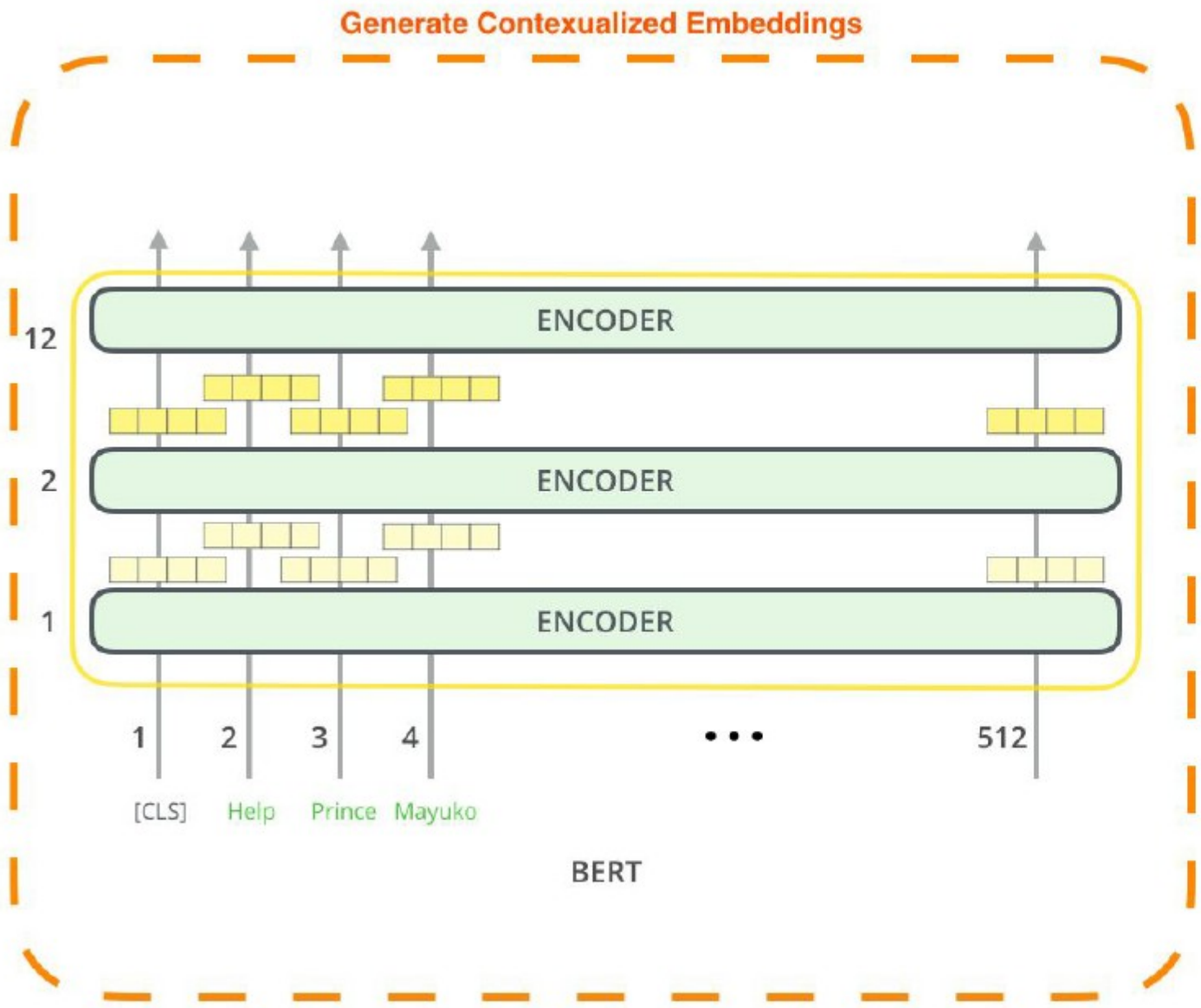


1 **Артур Кадурин, преподаватель**

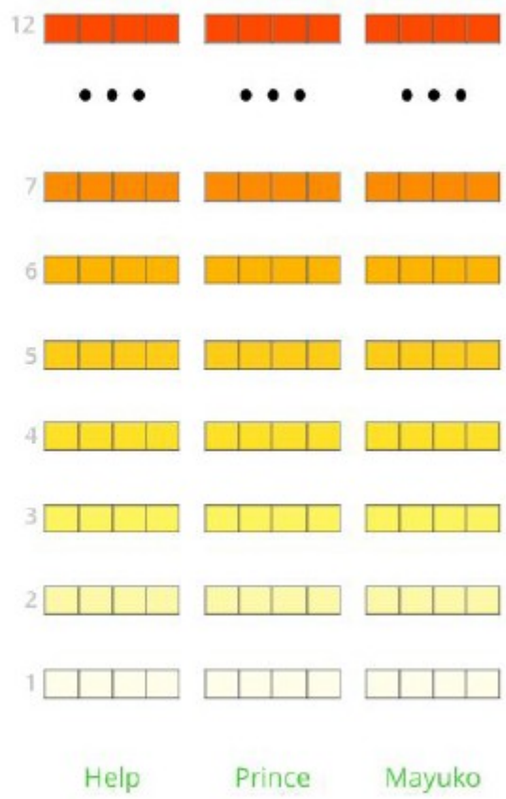
Similar to CNN concept!



BERT for feature extraction



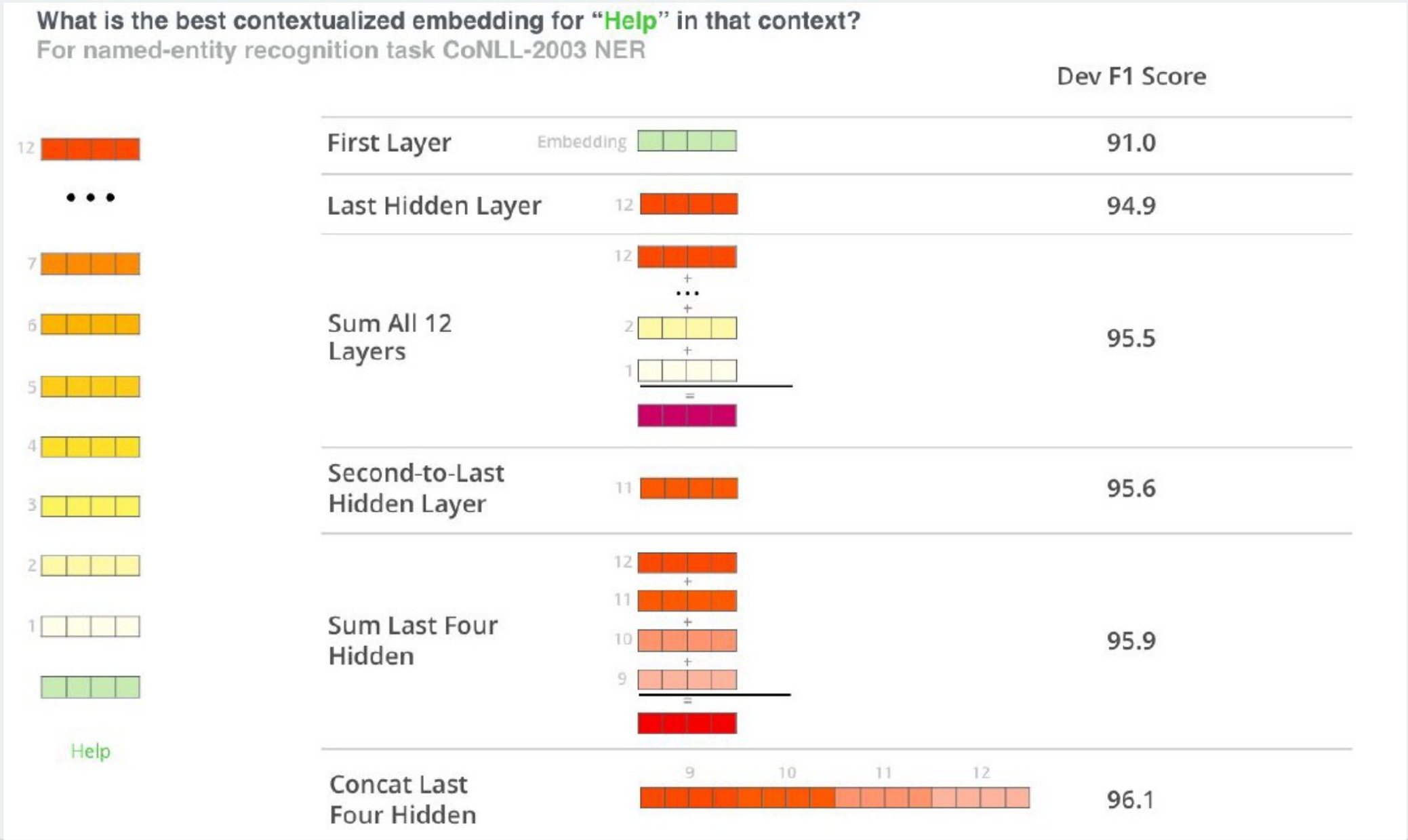
The output of each encoder layer along each token's path can be used as a feature representing that token.



But which one should we use?

Артур Кадурин, преподаватель

BERT for feature extraction



1 **Артур Кадурин, преподаватель**

SOTA: GPT-2

1 **Артур Кадурин, преподаватель**

SOTA: GPT-3

1 **Артур Кадурин, преподаватель**

175 billion parameters ¯_(ツ)_/¯

1 **Артур Кадурин, преподаватель**

GPT-2

- Transformer-based architecture
- trained to predict the **next** word
- 1.5 billion parameters
- Trained on 8 million web-pages

GPT-2

- Transformer-based architecture
- trained to predict the **next** word
- 1.5 billion parameters
- Trained on 8 million web-pages

On language tasks (question answering, reading comprehension, summarization, translation) works well **WITHOUT** fine-tuning

GPT-2: question answering

EXAMPLES

Who wrote the book the origin of species?

Correct answer: *Charles Darwin*
Model answer: `Charles Darwin`

What is the largest state in the U.S. by land mass?

Correct answer: *Alaska*
Model answer: `California`

GPT-2: fake news and hype

New AI fake text generator may be too dangerous to ... - The Guardian
https://www.theguardian.com/.../elon-musk-backed-ai-writes-convincing-news-fiction
4 days ago - The Elon Musk-backed nonprofit company OpenAI declines to release research publicly for fear of misuse. The creators of a revolutionary AI system that can write news stories and works of fiction – dubbed "deepfakes for text" – have taken the unusual step of not releasing ...

OpenAI built a text generator so good, it's considered too dangerous to ...
https://techcrunch.com/2019/02/17/openai-text-generator-dangerous/
12 hours ago - A storm is brewing over a new language model, built by non-profit artificial intelligence research company OpenAI, which it says is so good at ...

The AI Text Generator That's Too Dangerous to Make Public | WIRED
https://www.wired.com/story/ai-text-generator-too-dangerous-to-make-public/
4 days ago - In 2015, car-and-rocket man Elon Musk joined with influential startup backer Sam Altman to put artificial intelligence on a new, more open ...

Elon Musk-backed AI Company Claims It Made a Text Generator ...
https://gizmodo.com/elon-musk-backed-ai-company-claims-it-made-a-text-gener-183...
Elon Musk-backed AI Company Claims It Made a Text Generator That's **Too Dangerous to** Release · Rhett Jones · Friday 12:15pm · Filed to: OpenAI Filed to: ...

Scientists have made an AI that they think is too dangerous to ...
https://www.weforum.org/.../amazing-new-ai-churns-out-coherent-paragraphs-of-text/
3 days ago - Sample outputs suggest that the AI system is an extraordinary step forward, producing text rich with context, nuance and even something ...

New AI Fake Text Generator May Be Too Dangerous To ... - Slashdot
https://news.slashdot.org/.../new-ai-fake-text-generator-may-be-too-dangerous-to-rele...
3 days ago - An anonymous reader shares a report: The creators of a revolutionary AI system that can write news stories and works of fiction -- dubbed ...

Top stories

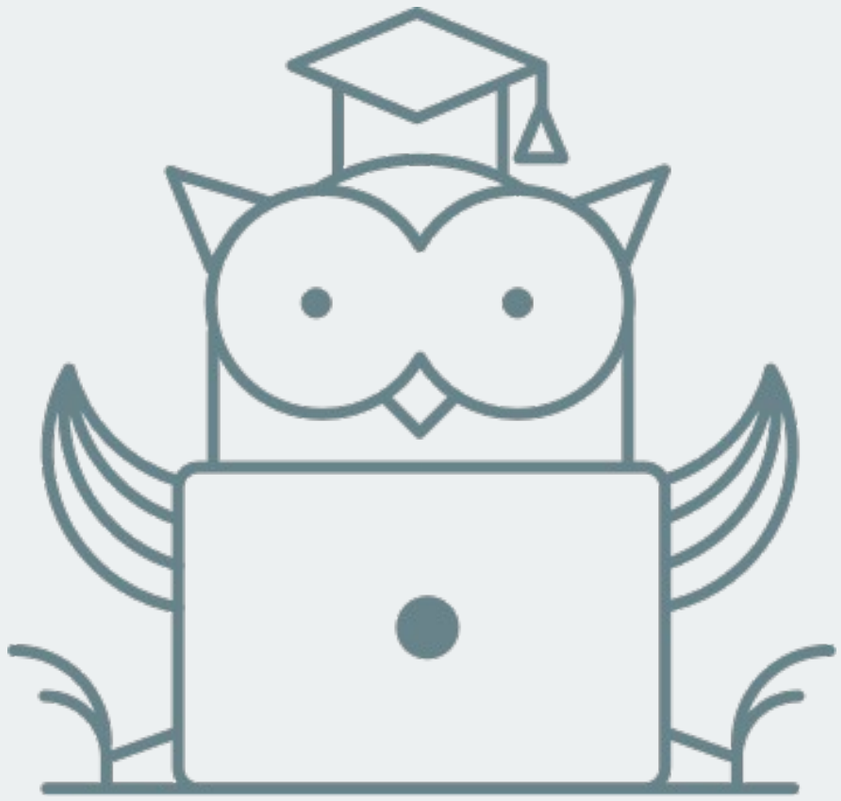
OpenAI built a text generator so good, it's considered too dangerous to release
TechCrunch
11 hours ago

Elon Musk's AI company created a fake news generator it's too scared to make public
BGR.com
9 hours ago

The AI That Can Write A Fake News Story From A Handful Of Words
NDTV.com
2 hours ago

When Is Technology Too Dangerous to Release to the Public?
Slate • 2 days ago

Scientists Developed an AI So Advanced They Say It's Too Dangerous to Release
ScienceAlert • 6 days ago



Спасибо
за внимание!