

Introduction au Machine Learning

Transformers

Emmanuel Dellandrea - emmanuel.dellandrea@ec-lyon.fr

Natural Language Processing



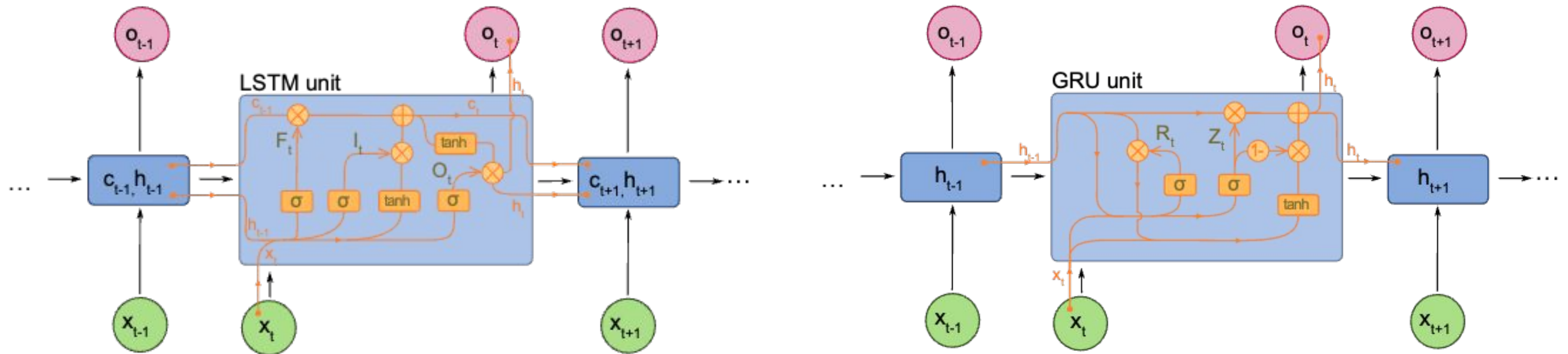
Objective: Model the syntactic and grammatical structures of a language.

Numerous applications:

- Speech recognition
- Machine translation
- Text comprehension
- Text generation (e.g., Q/A or summaries)
- Paraphrase detection
- Sentiment analysis

Language models

Main models until 2017: Recurrent neural networks like LSTM and GRU



Limits of these models:

- Difficulties in processing long sequences
- Frequent overfitting issues
- Challenges in modeling complex relationships
- Computations that cannot be parallelized (need for the previous word)

Transformers

Language models relying on a **self-attention mechanism**

Published in 2017

→ **Revolution in the field of Natural Language Processing (NLP)**

arXiv:1706.03762v5 [cs.CL] 6 Dec 2017

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez*[†]
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Illia Polosukhin*[‡]
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

1 Introduction

Recurrent neural networks, long short-term memory [13] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and

*Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Łukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

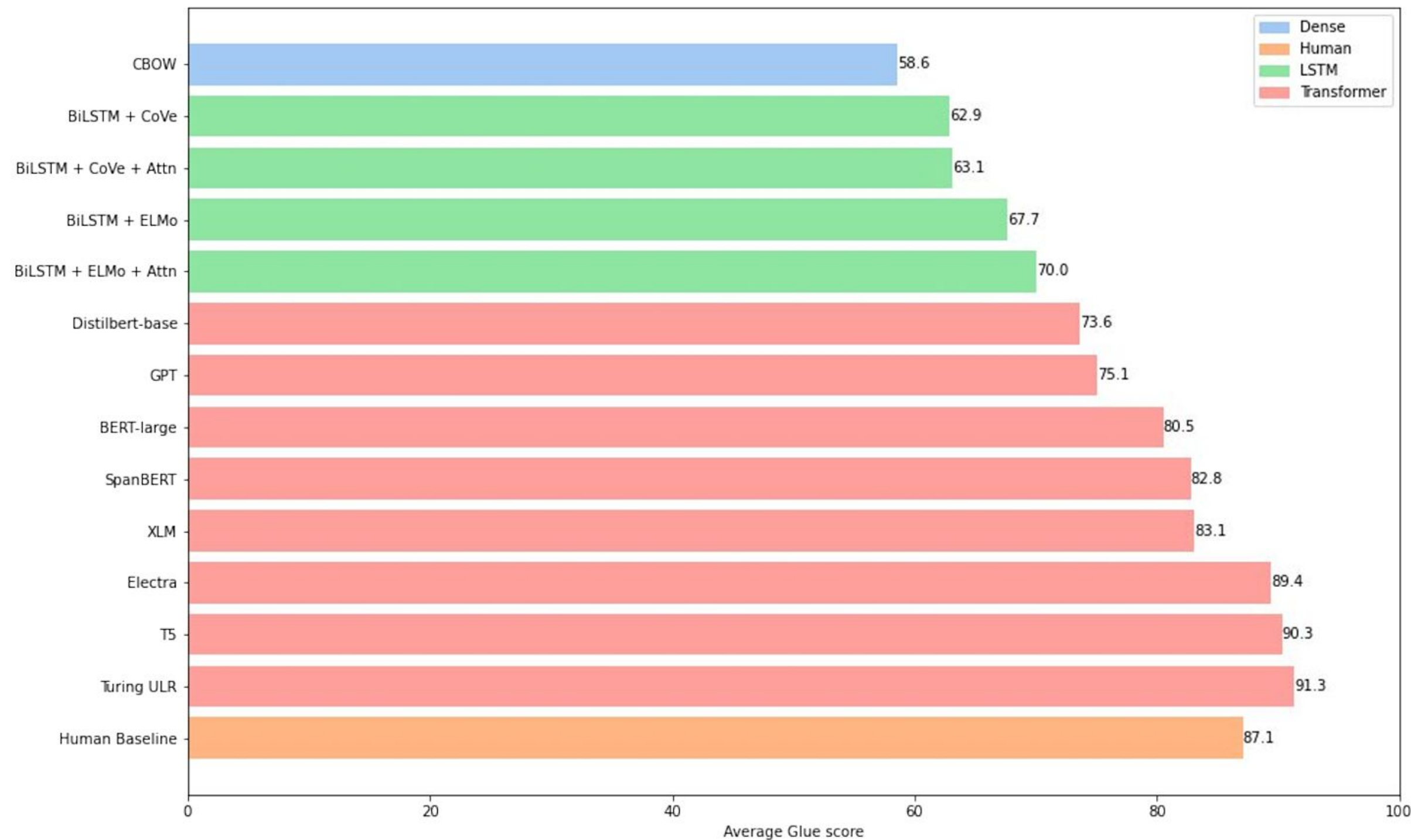
[†]Work performed while at Google Brain.

[‡]Work performed while at Google Research.

Objectives

- Process sequences (ideally the entire sentence)
- Easy to distribute on multiple GPUs
- Faster training than with RNN
- Initially for NLP tasks
- Allows to train huge models on gigantic datasets
- Allows for a pretraining session to pool trainings (at least partially) for multiple tasks

Evolution of performances



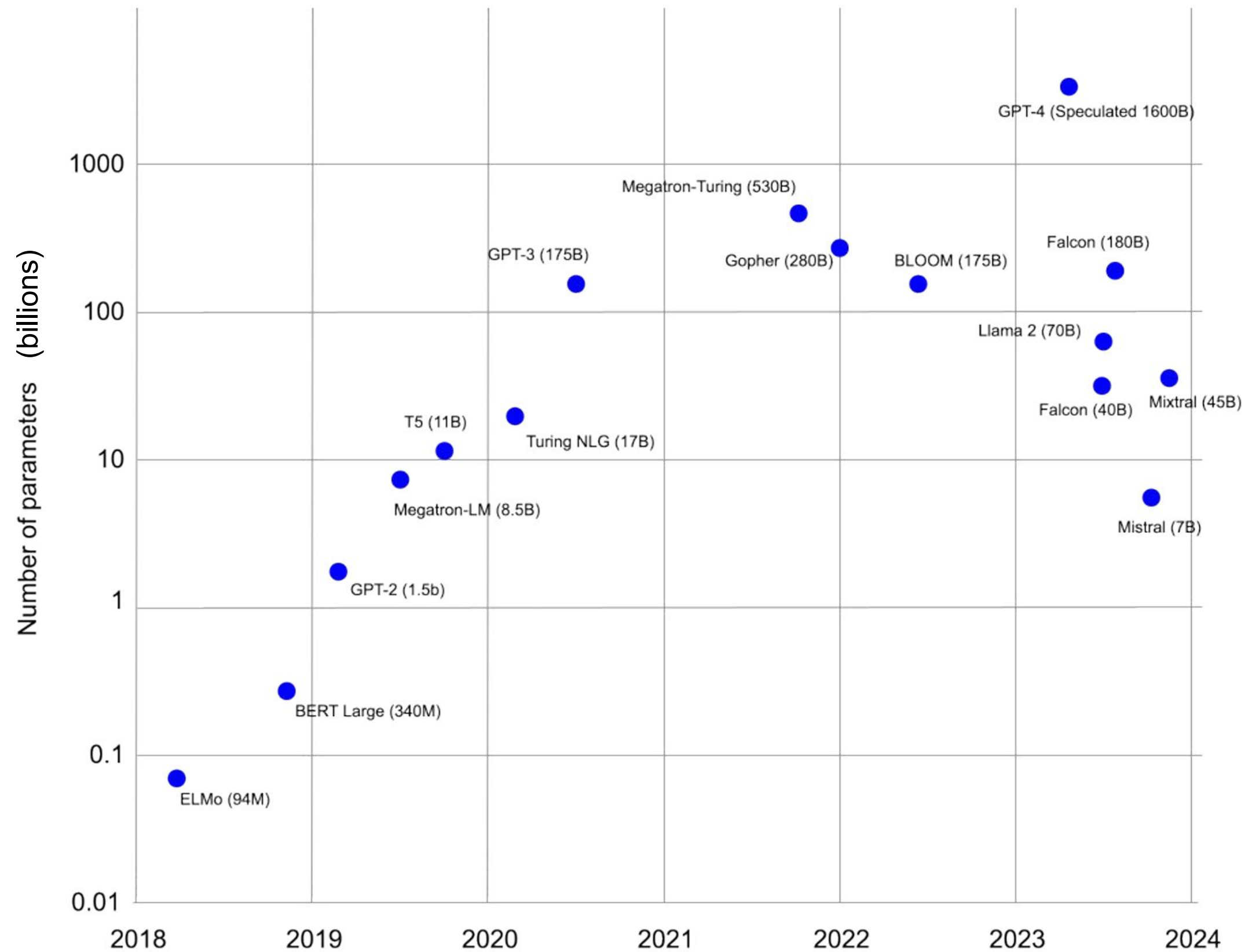
This article is more than 5 months old

ChatGPT better than undergraduates at solving SAT problems, study suggests

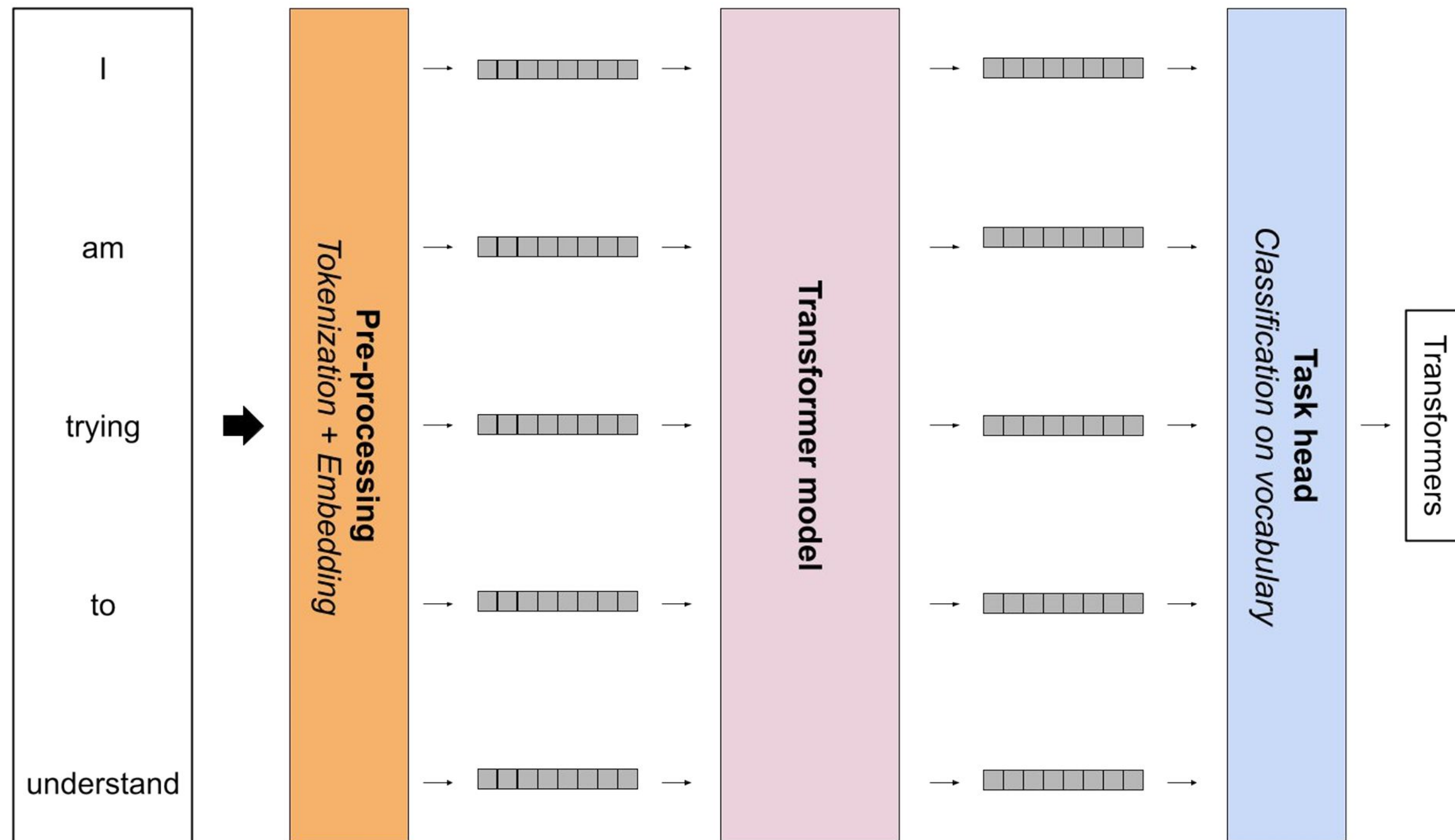
Researchers at UCLA found GPT-3 solved 80% of reasoning problems correctly compared with 60% of humans



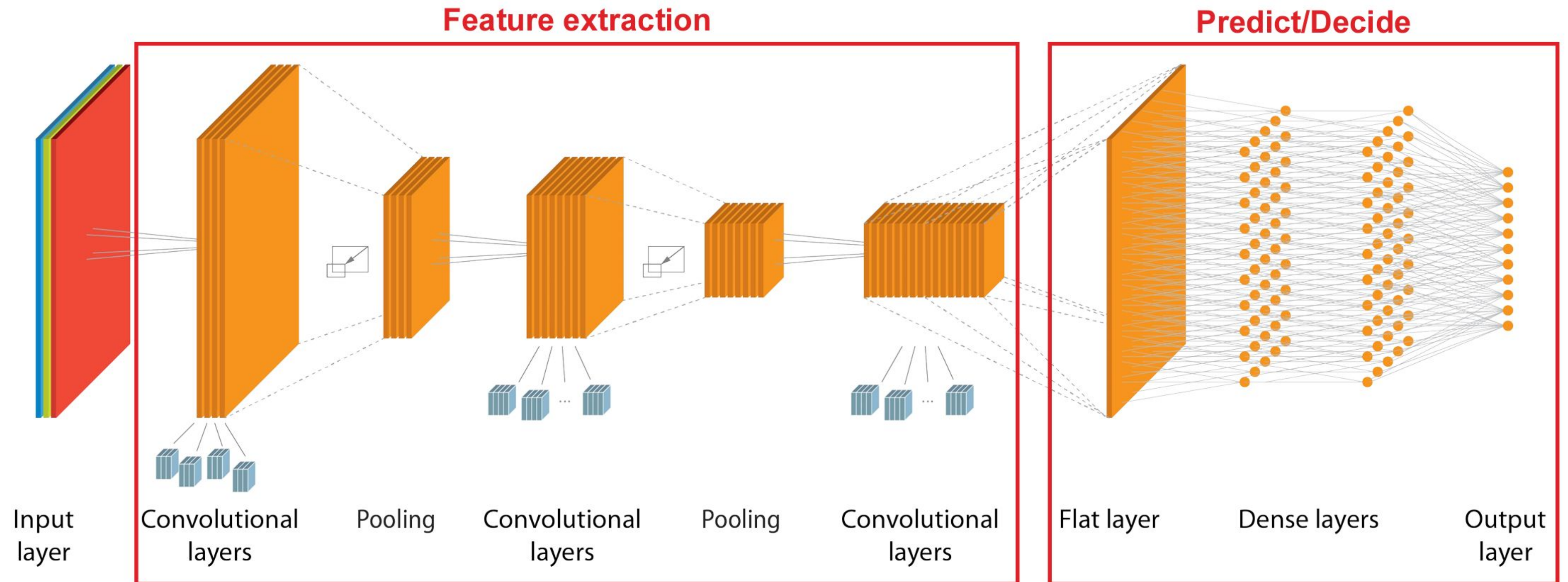
Size of Transformers



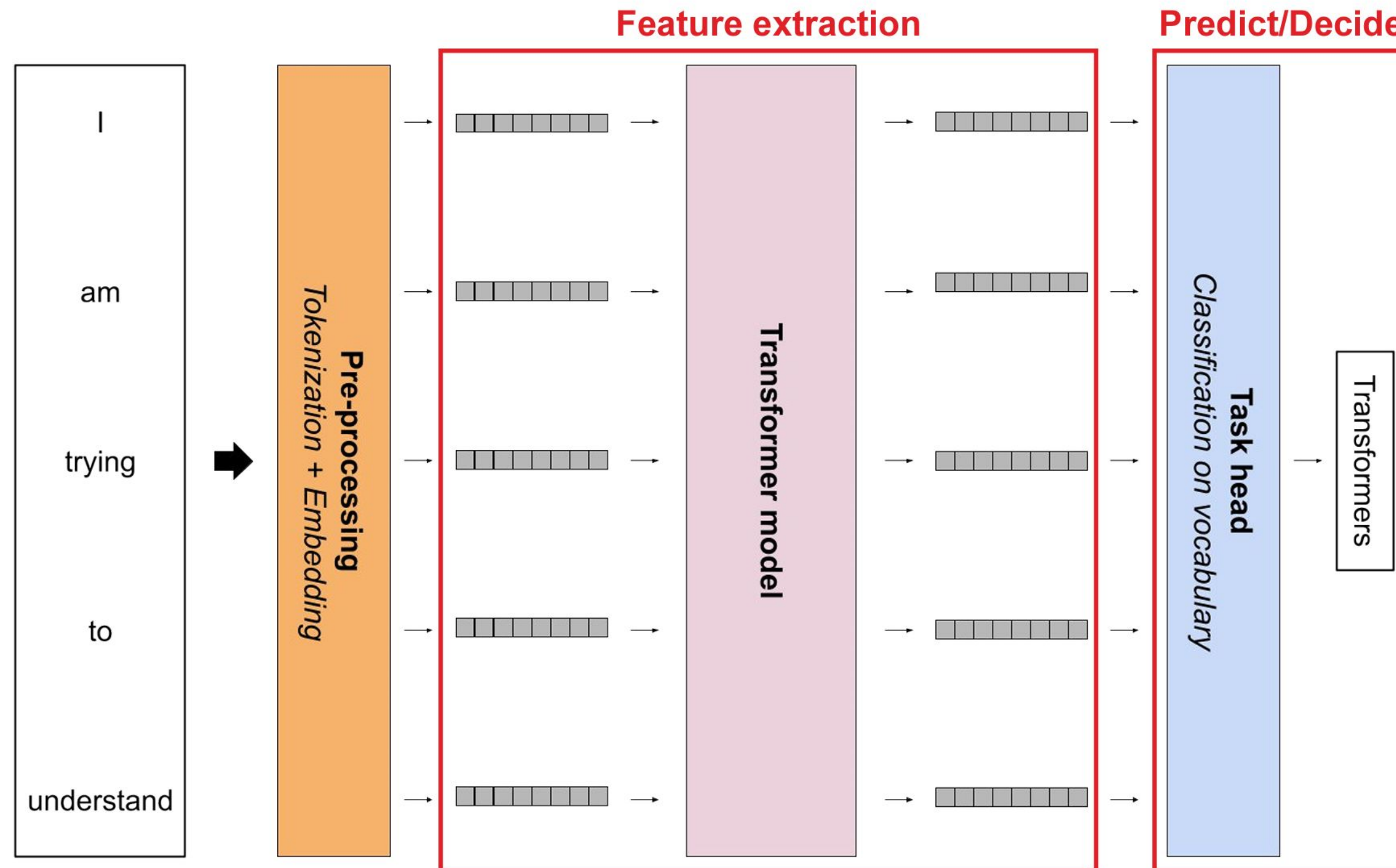
Example of NLP system



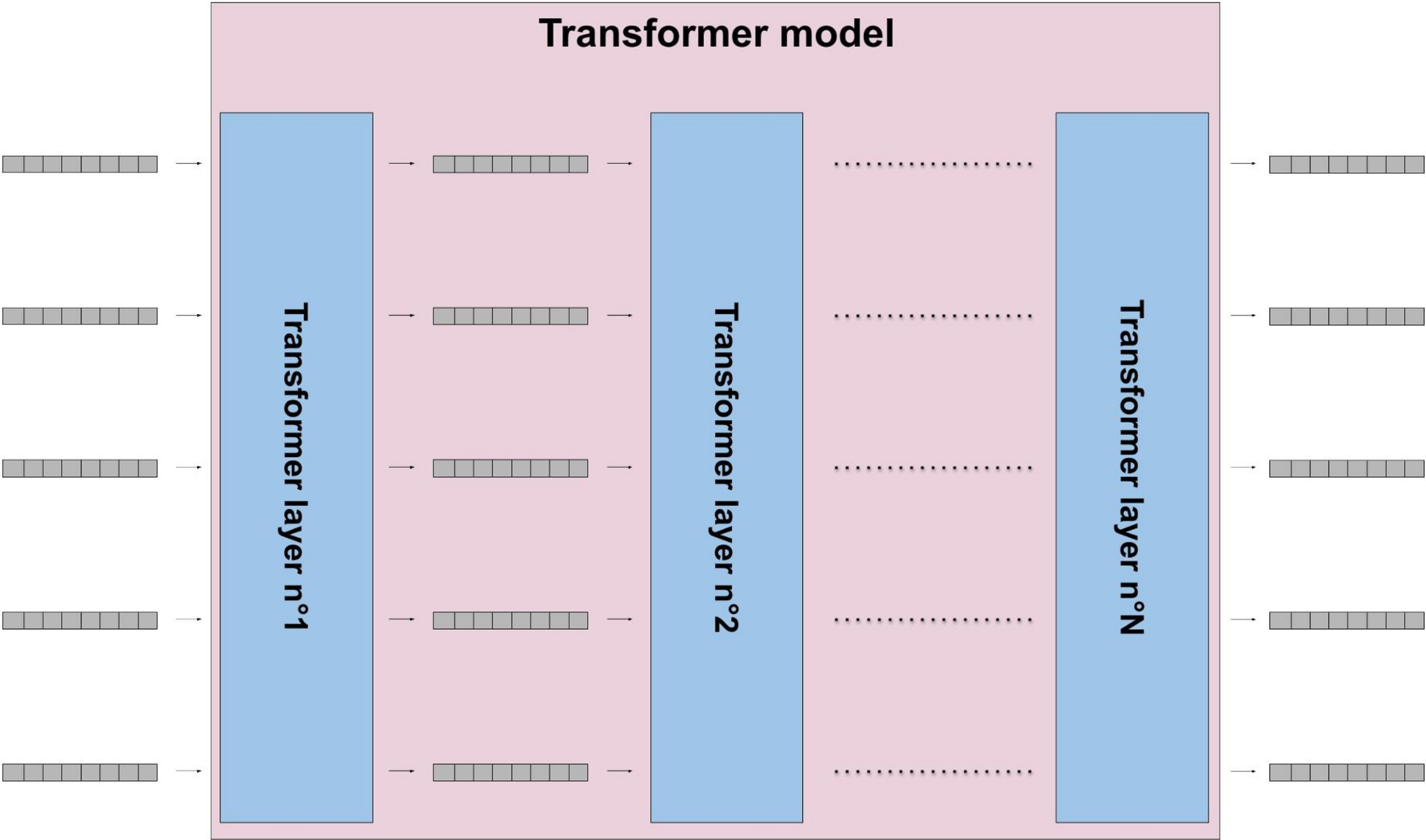
Feature extraction reminder



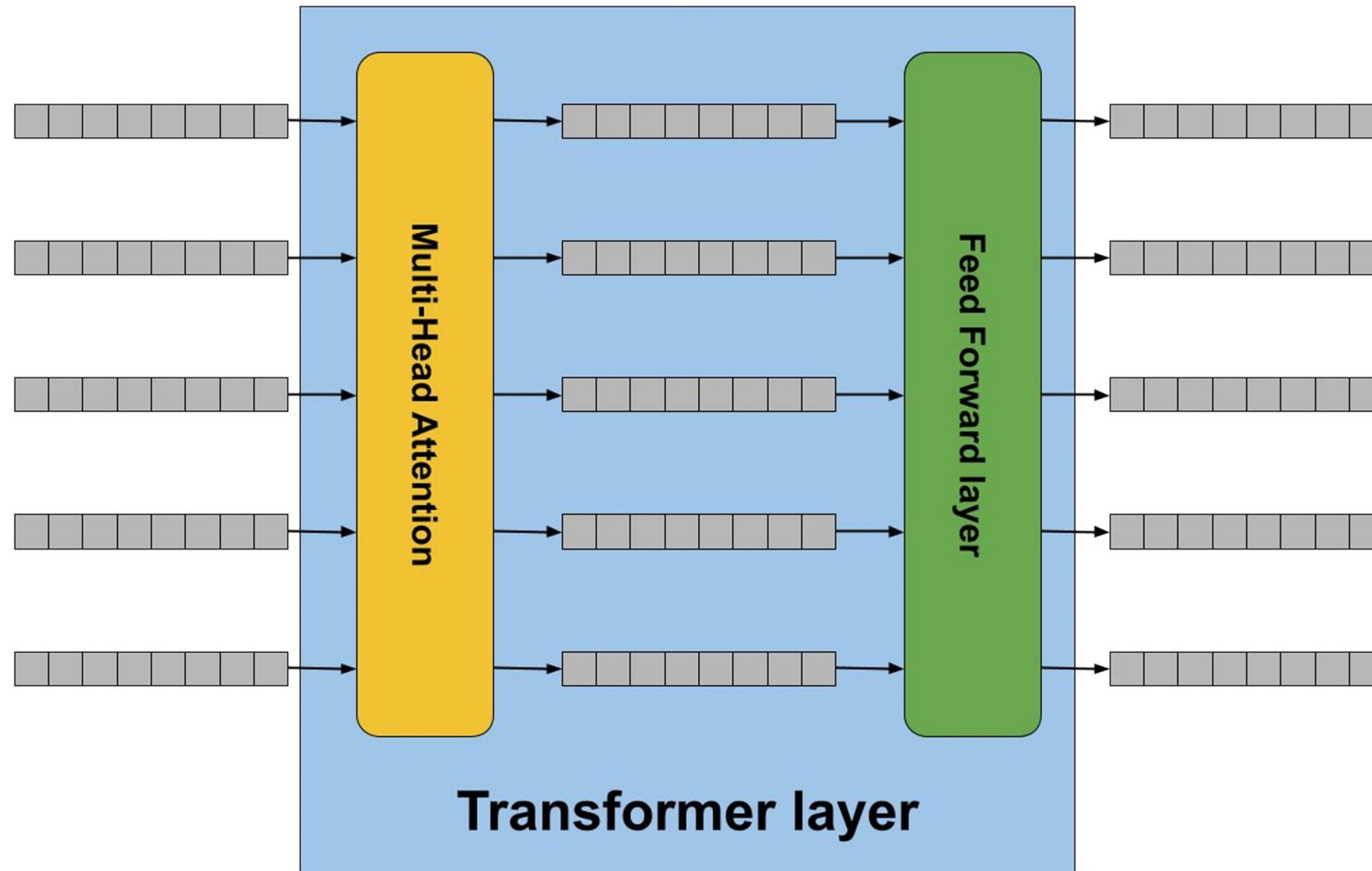
Feature extraction with Transformers



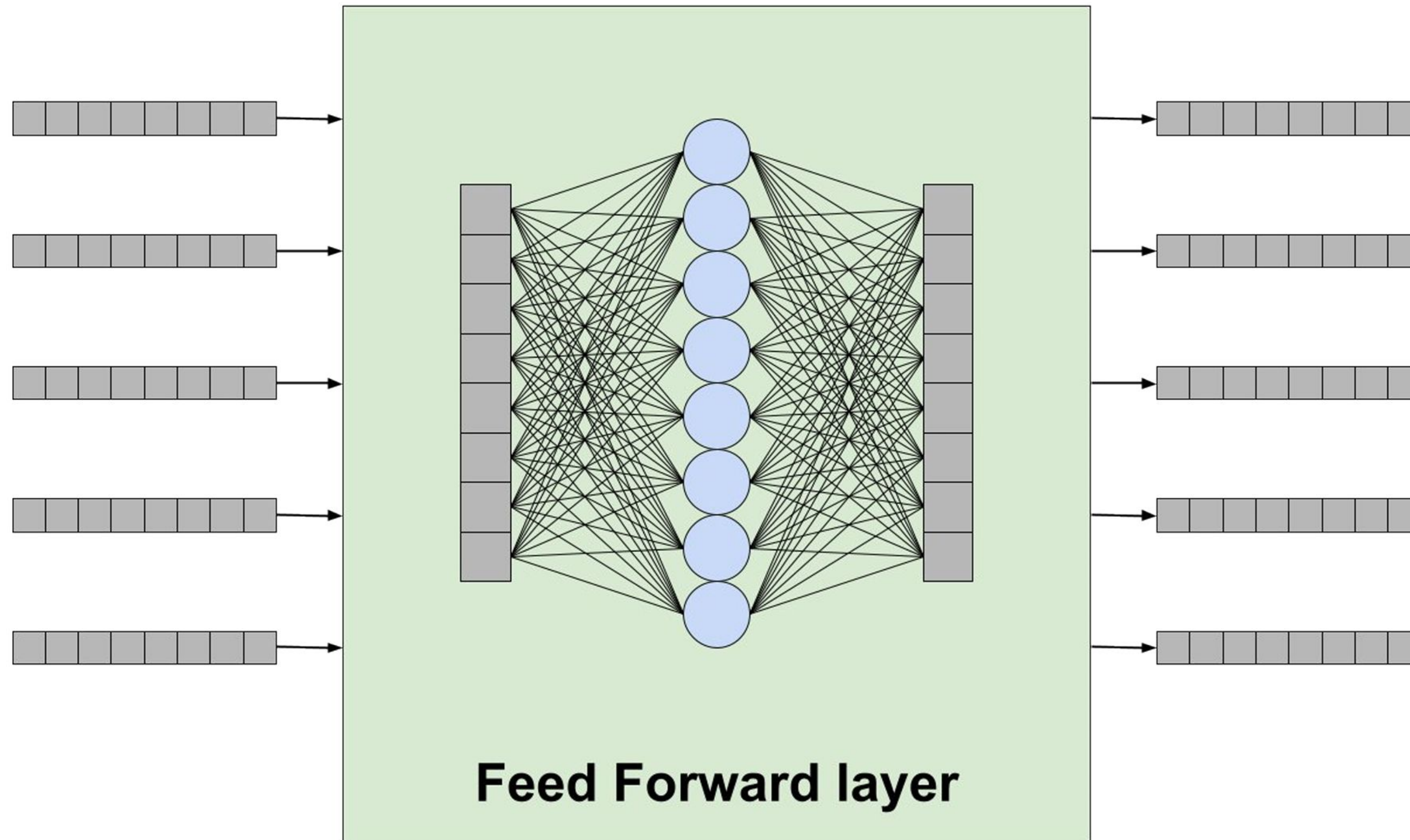
Typical Transformer Architecture



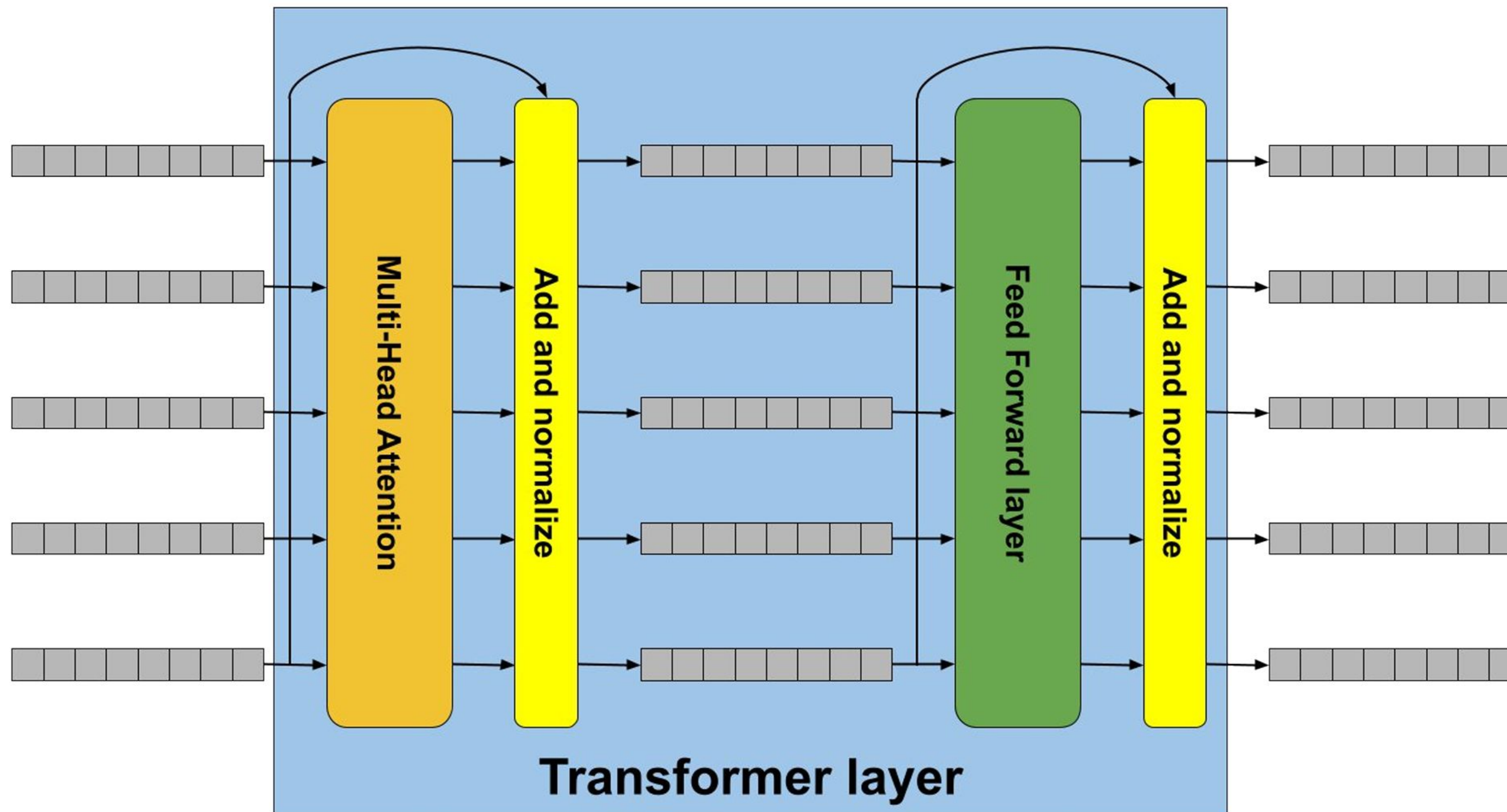
Simplified Transformer Layer



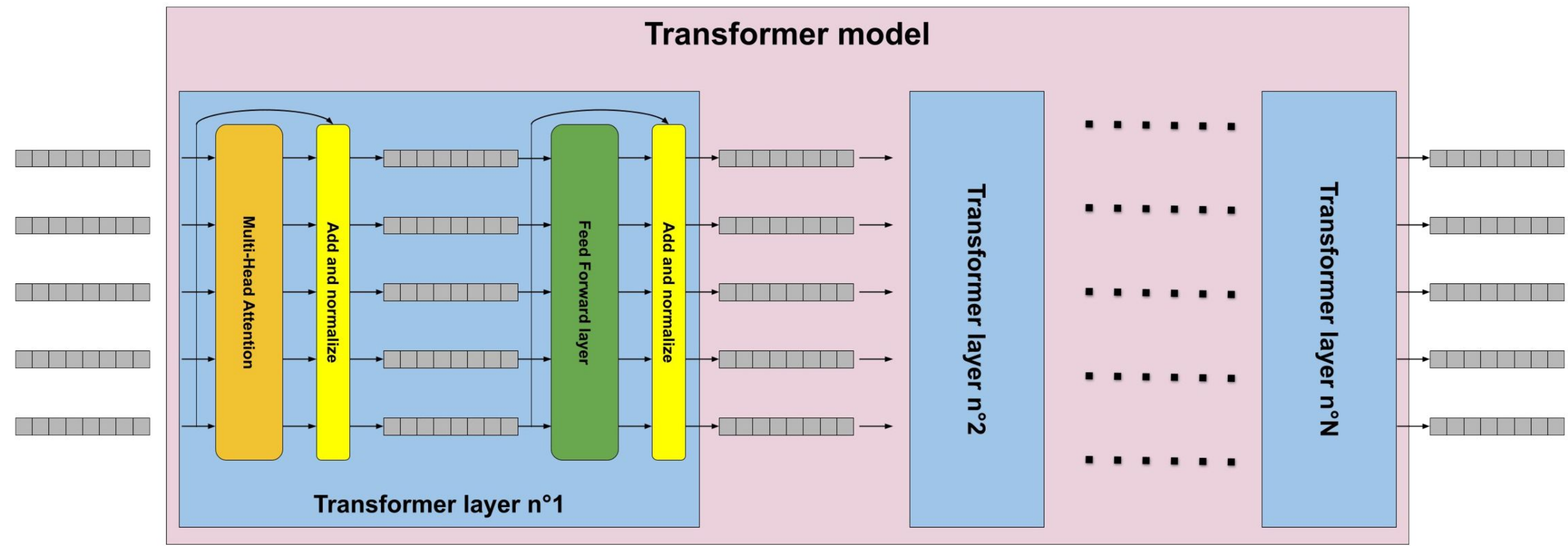
Feed Forward Layer



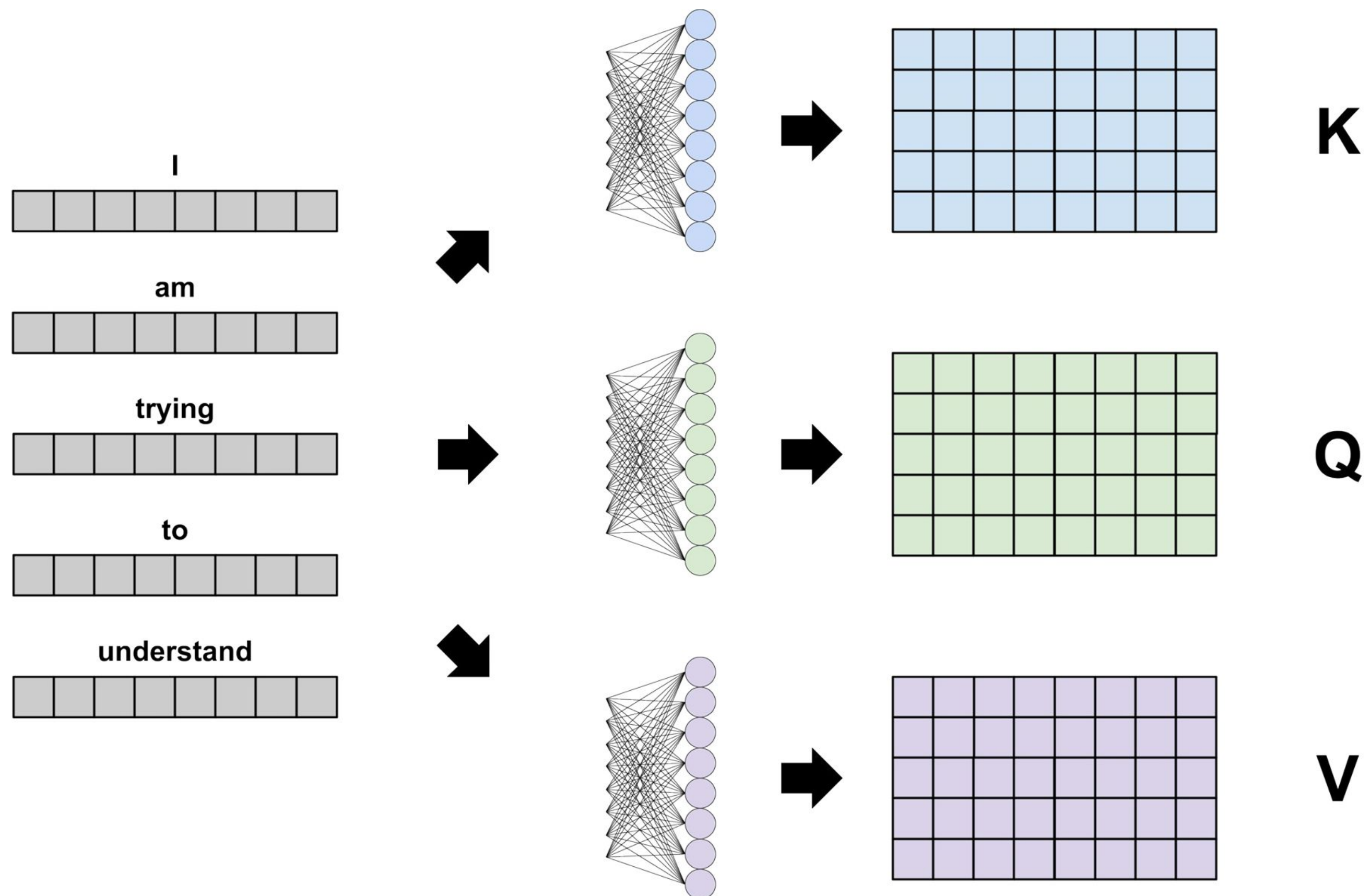
Transformer Layer



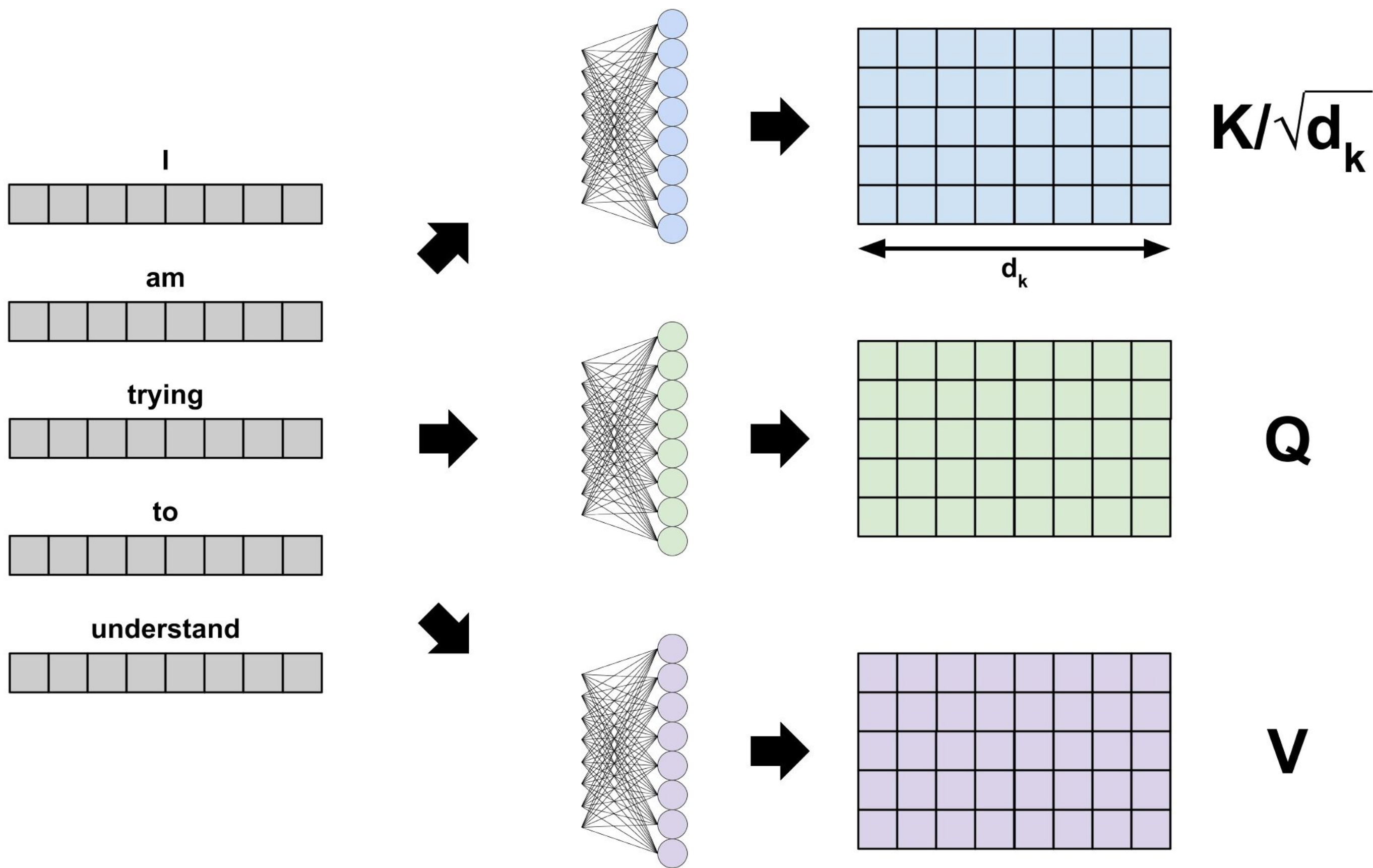
Transformer architecture summary



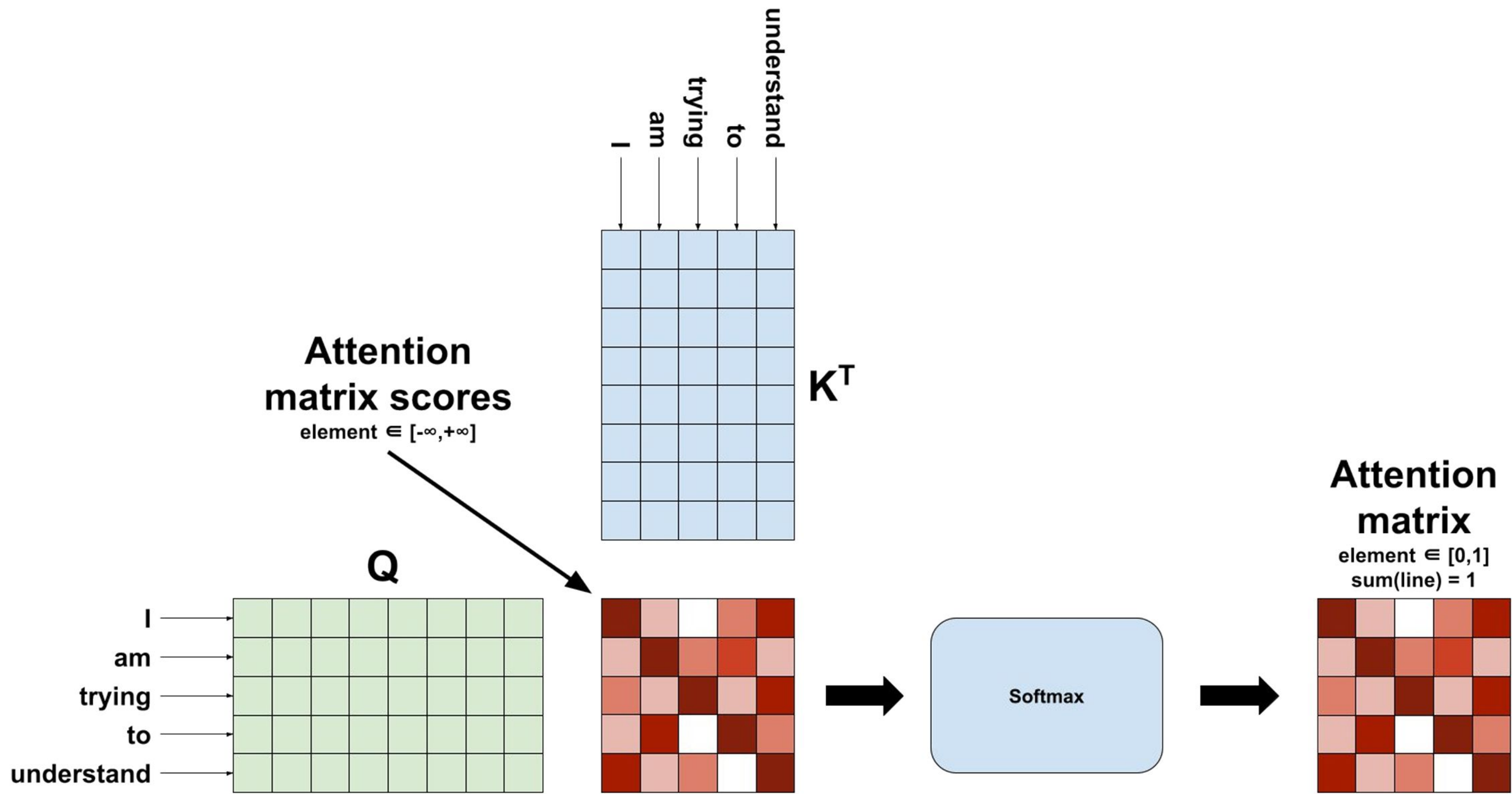
Attention mechanism



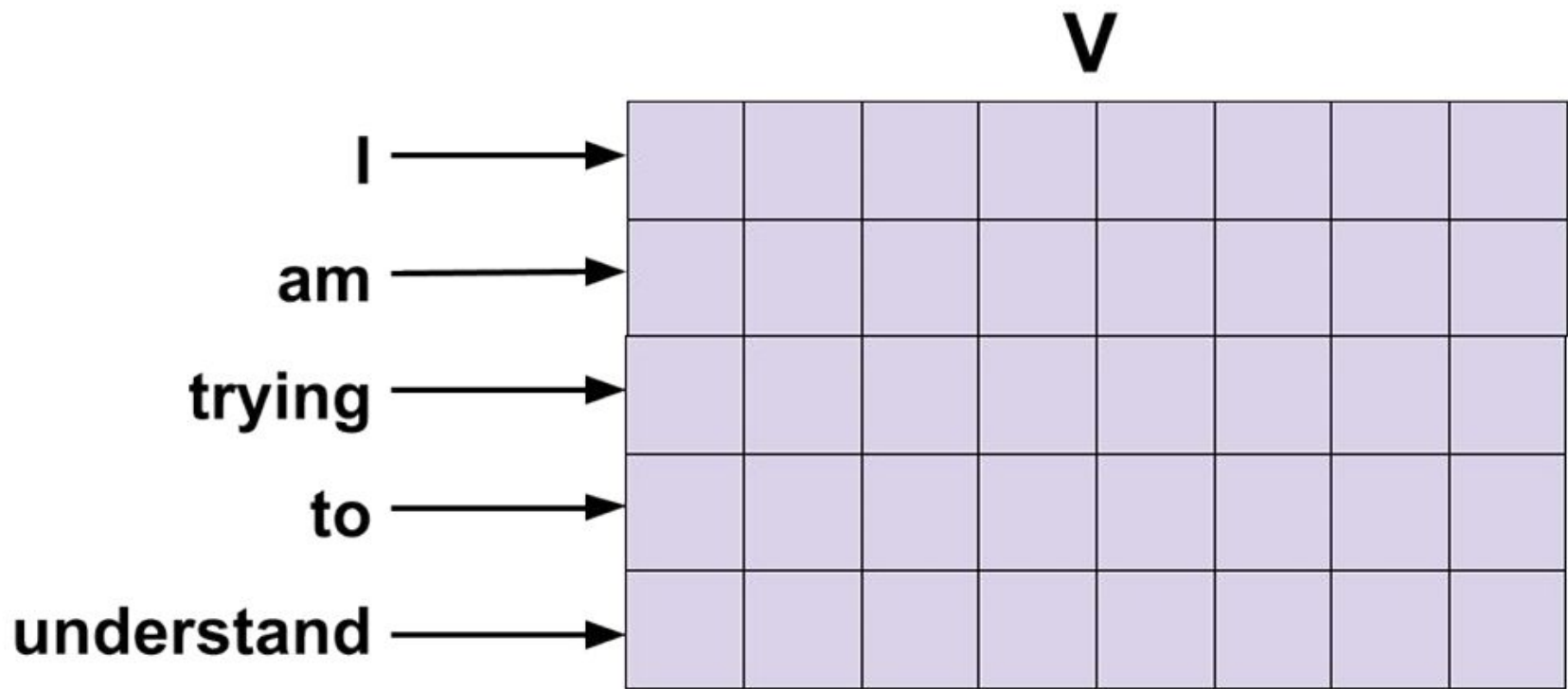
Attention mechanism



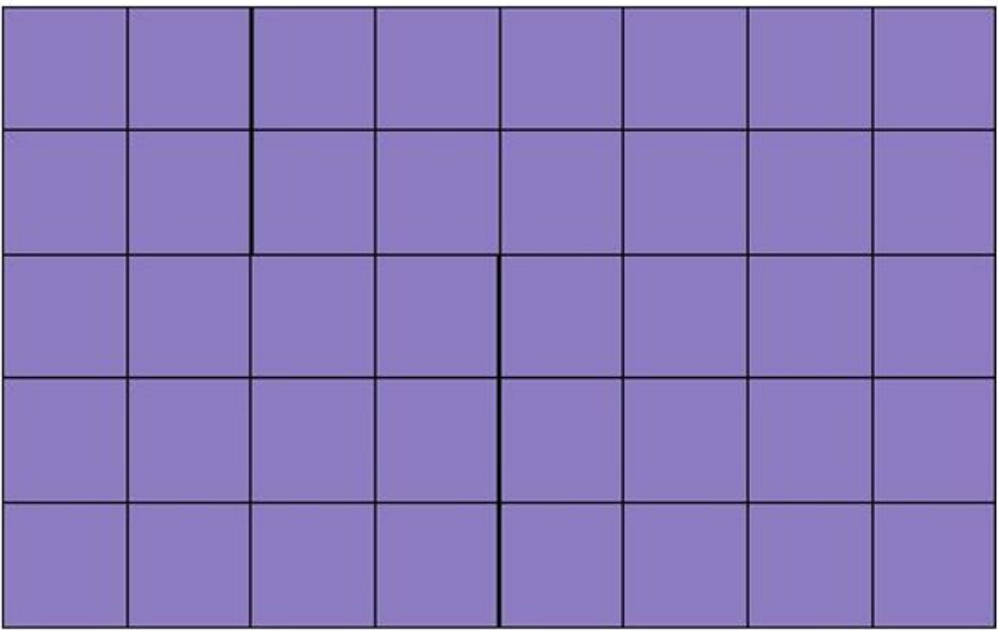
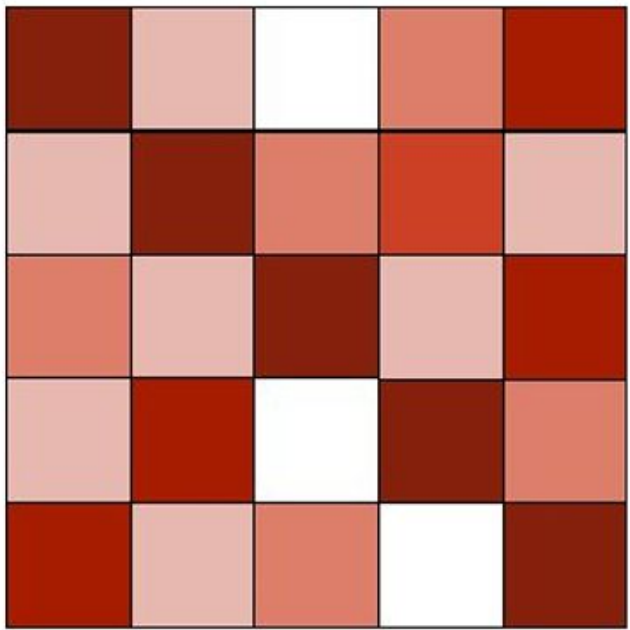
Attention explained



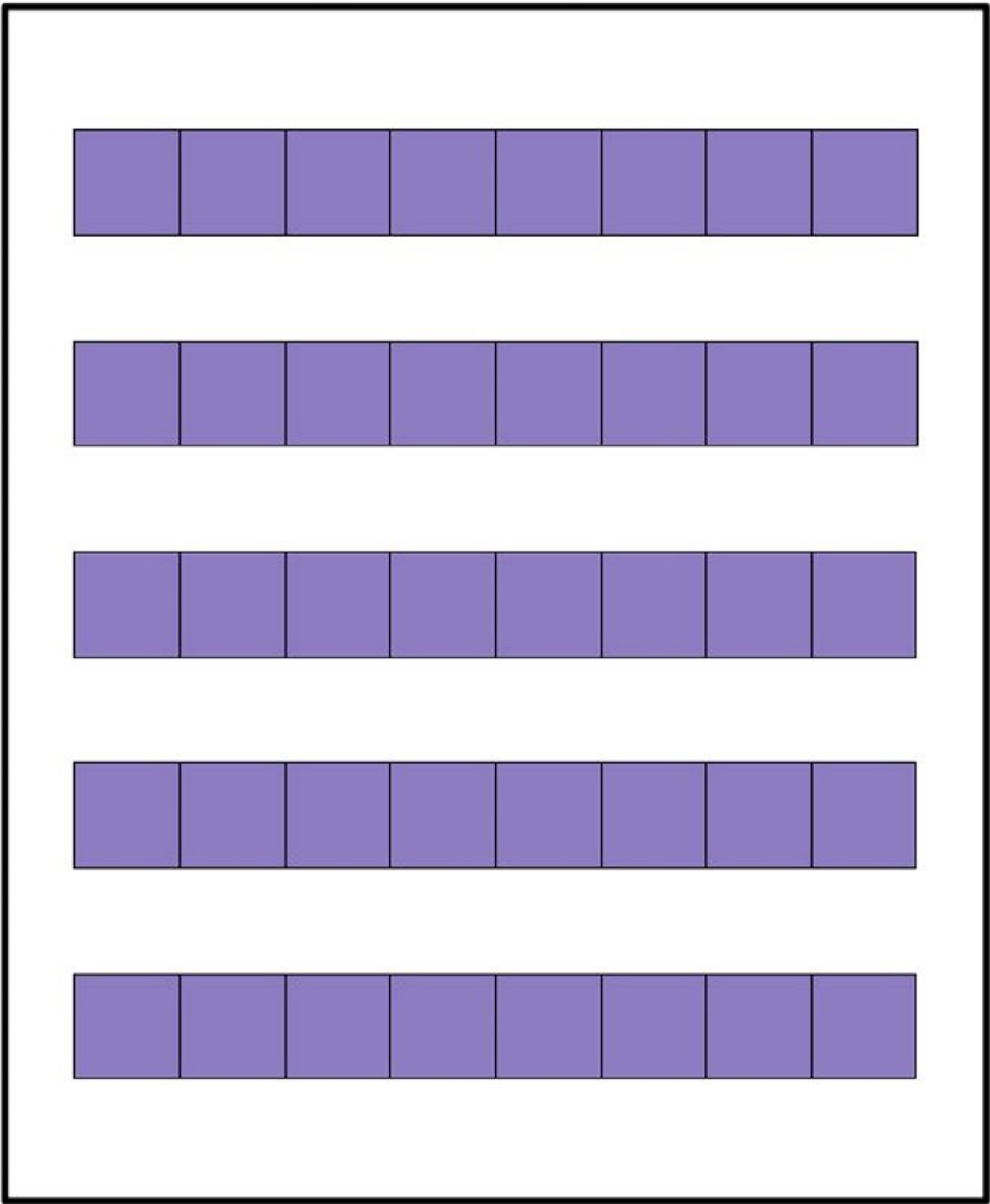
Attention explained



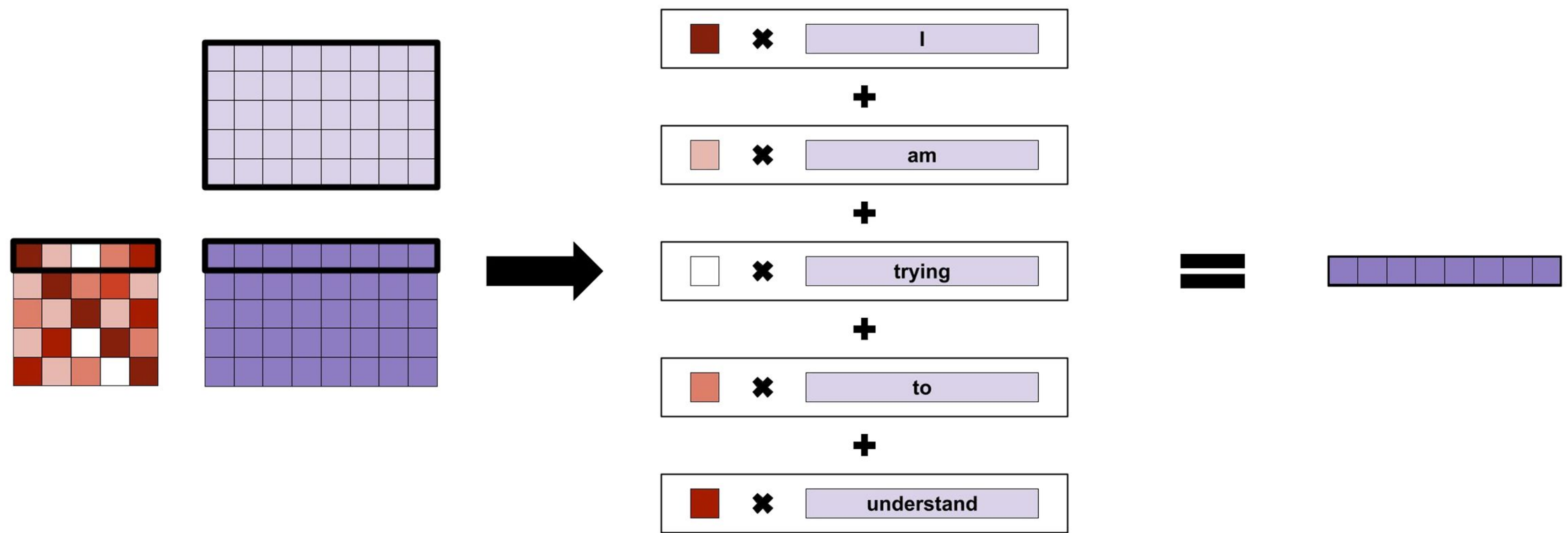
Attention matrix



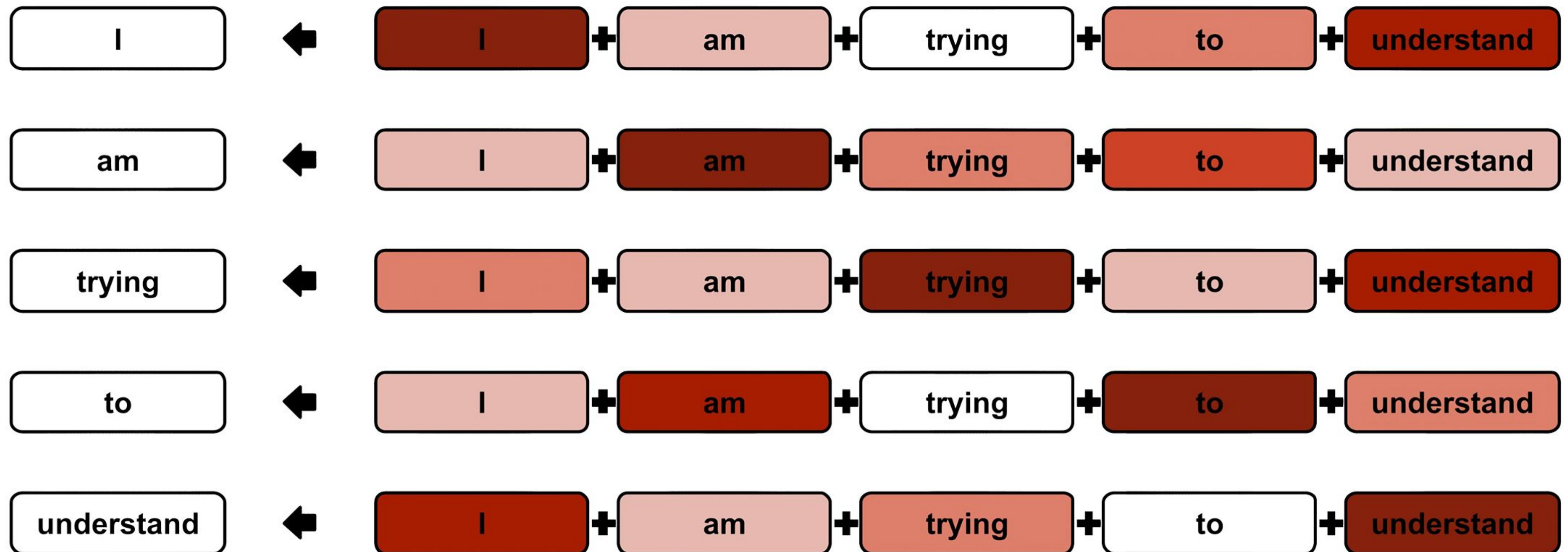
Output sequence



Attention explained



Intuition behind the Attention mechanism

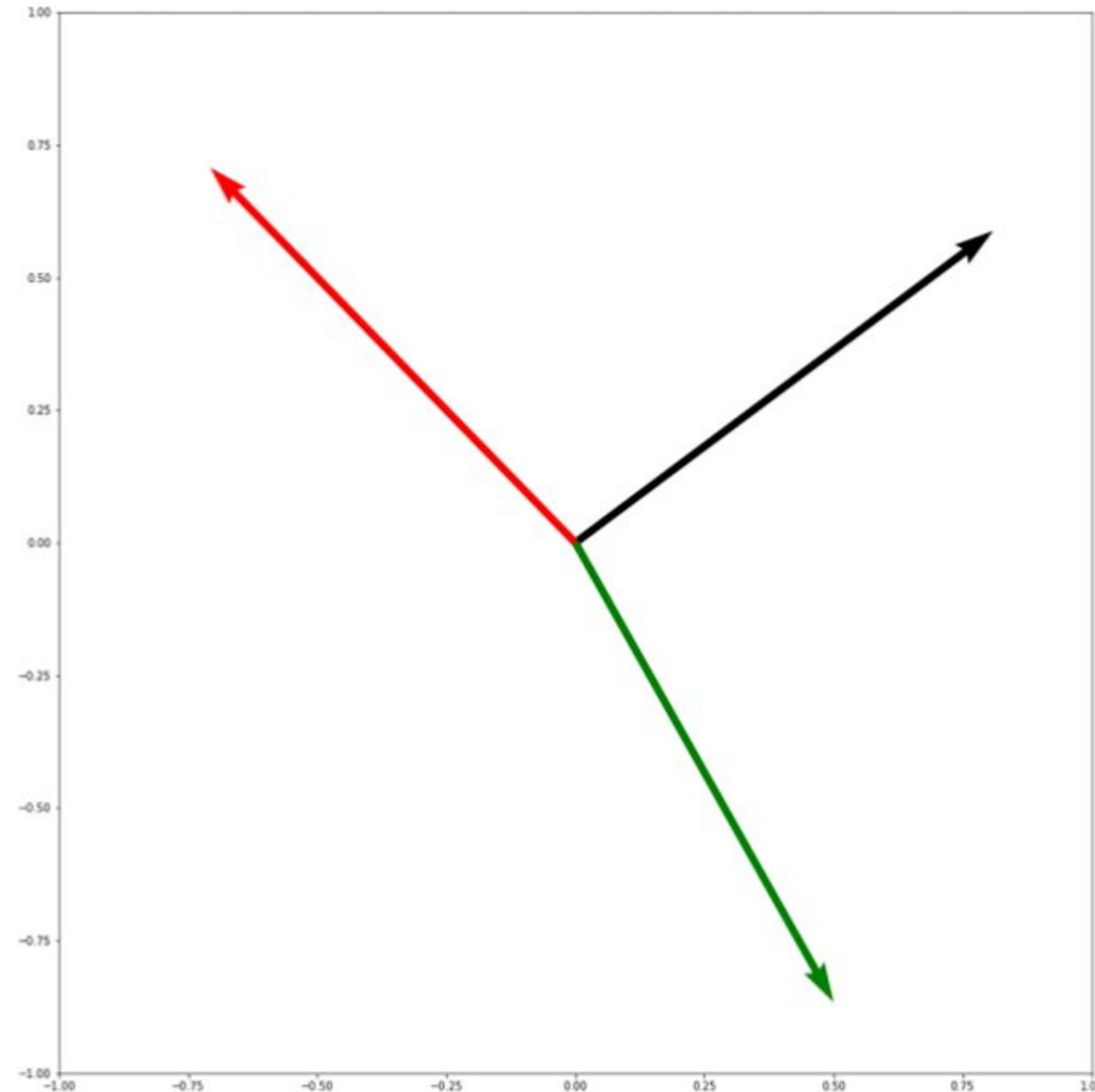


Intuition behind the Attention mechanism

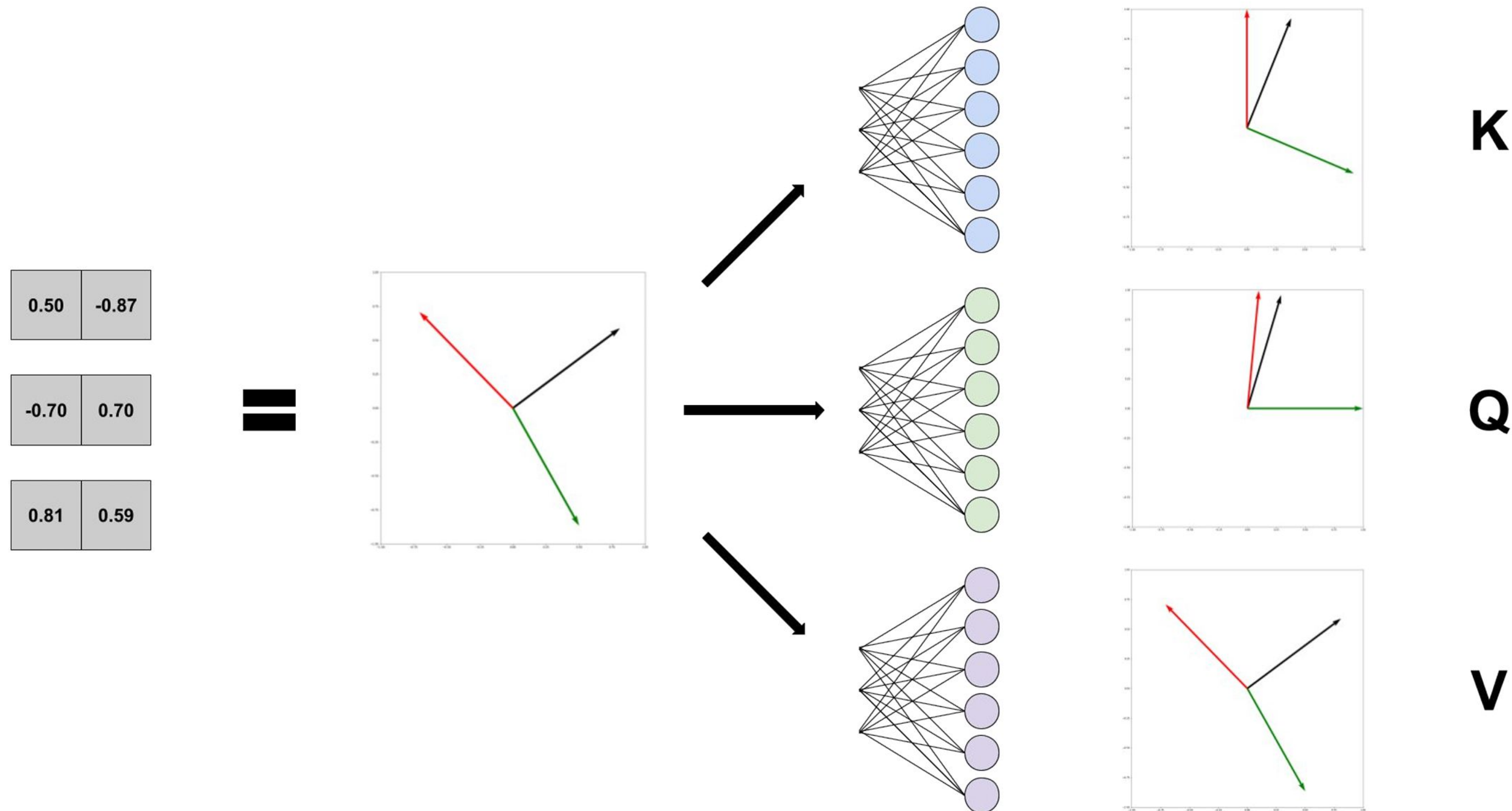
The big dog



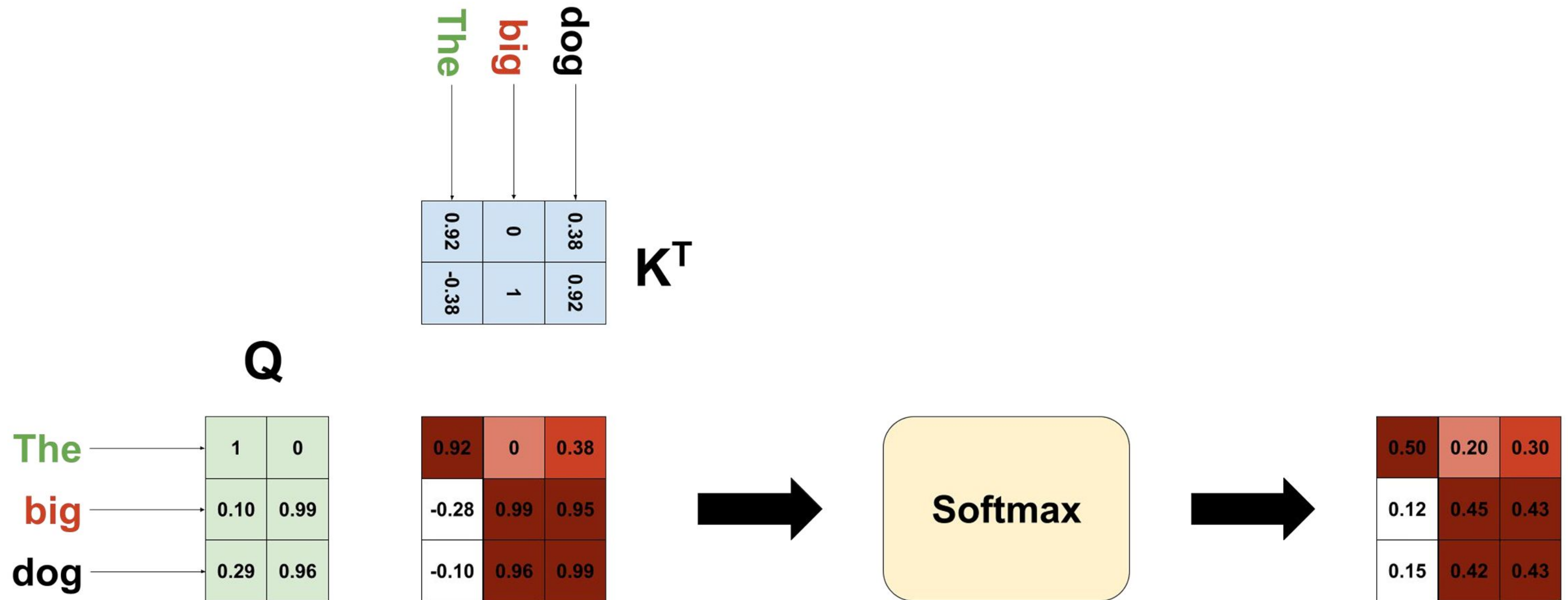
The : (0.50, -0.87)
big : (-0.70, 0.70)
dog : (0.81, 0.59)



Intuition behind the Attention mechanism



Intuition behind the Attention mechanism



Intuition behind the Attention mechanism

0.50	0.20	0.30
0.12	0.45	0.43
0.15	0.42	0.43

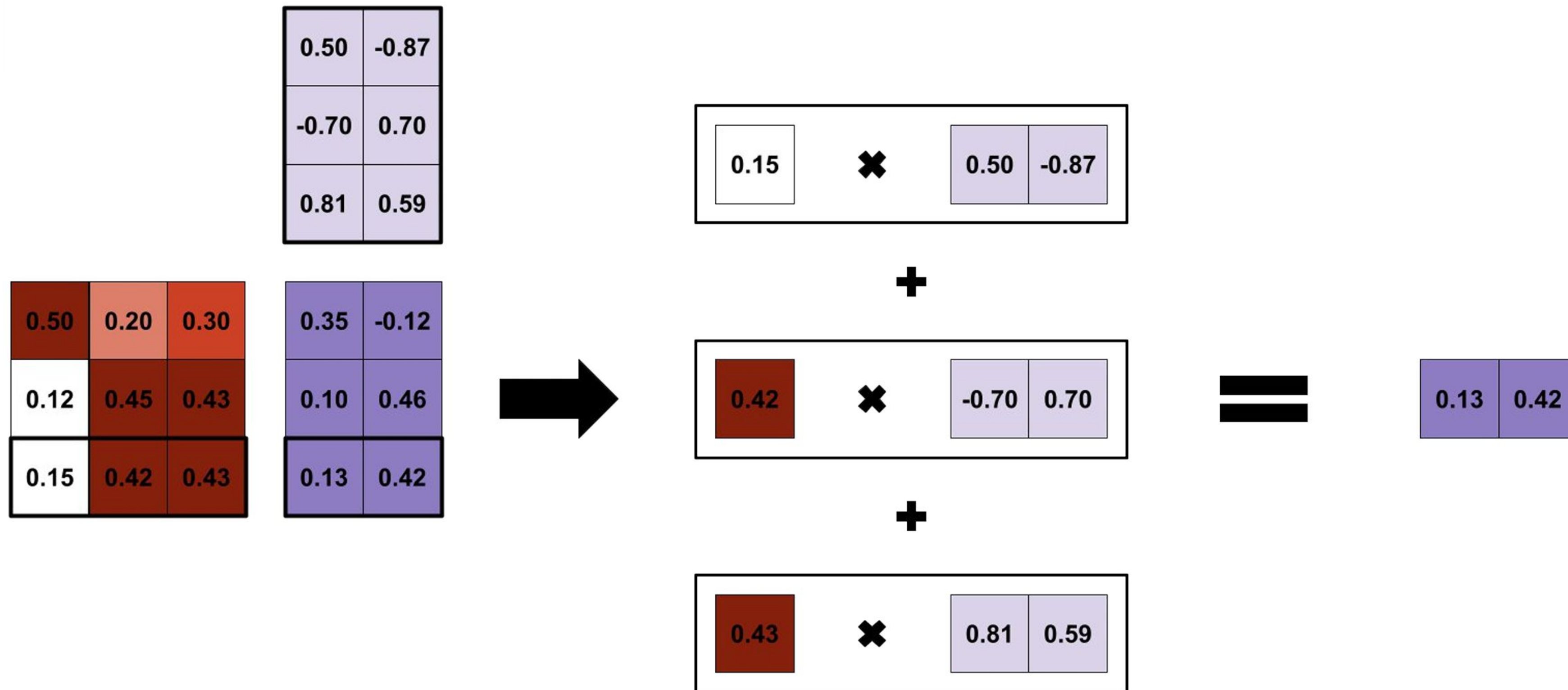


0.50	-0.87
-0.70	0.70
0.81	0.59

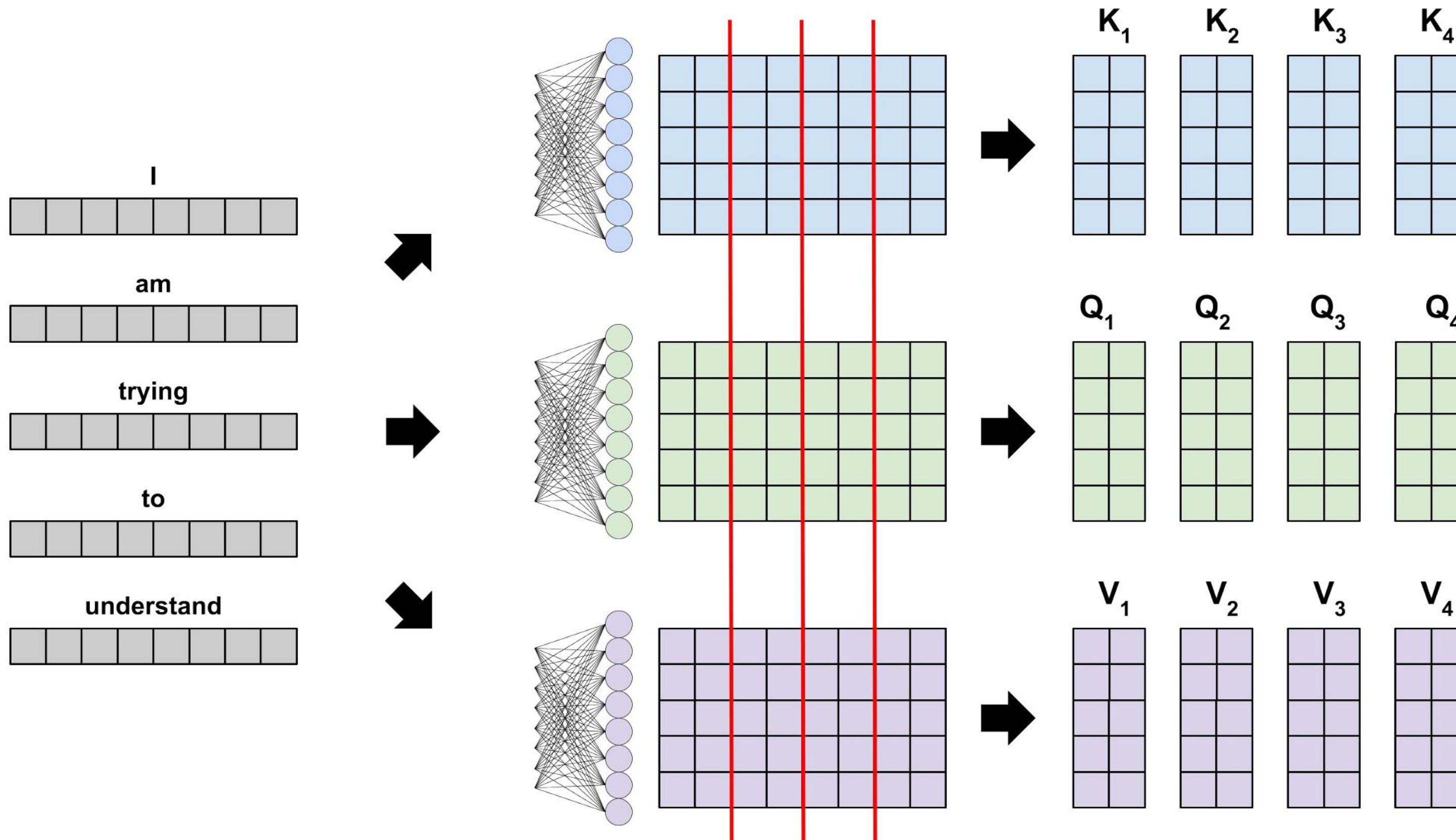


0.35	-0.12
0.10	0.46
0.13	0.42

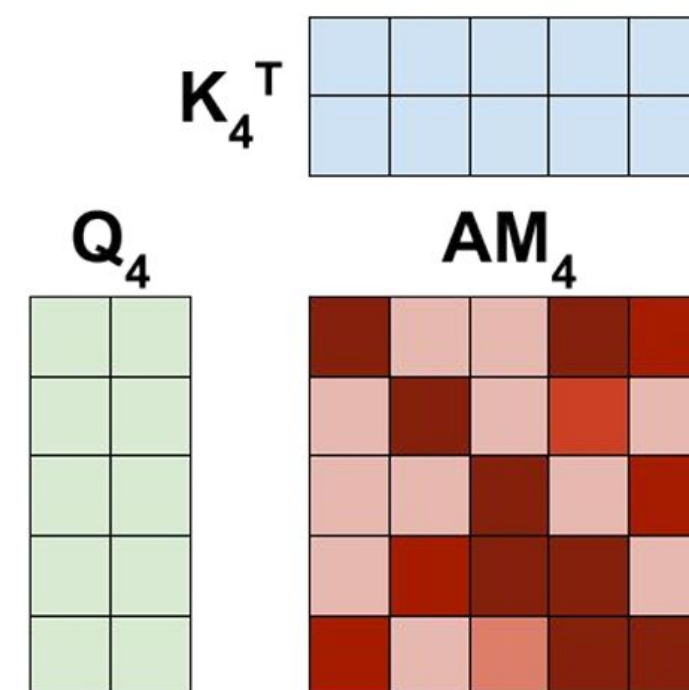
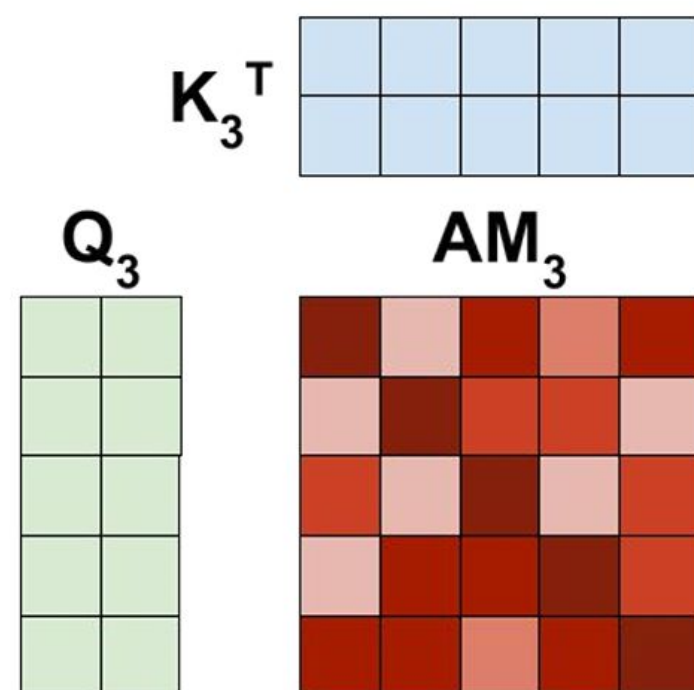
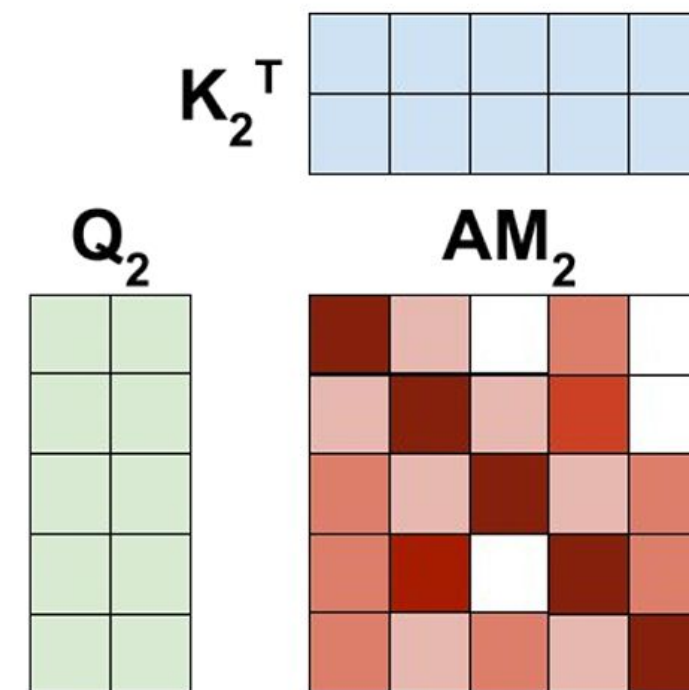
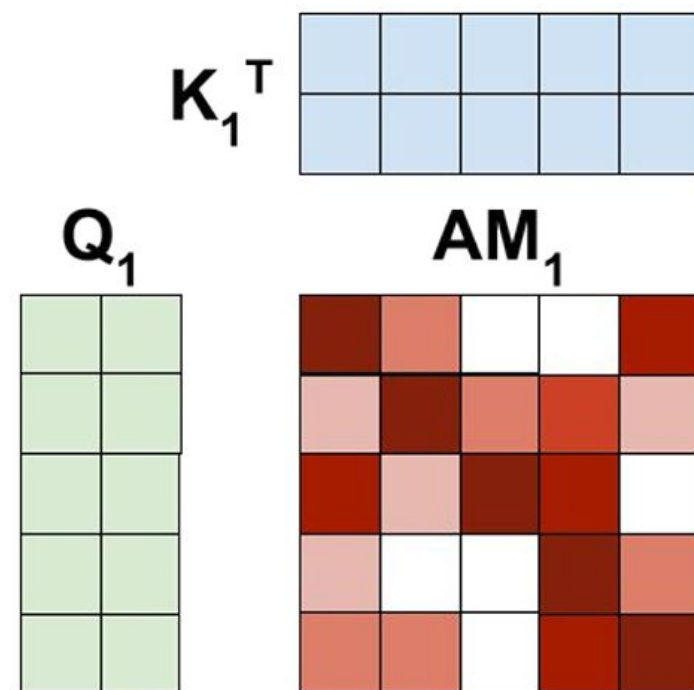
Intuition behind the Attention mechanism



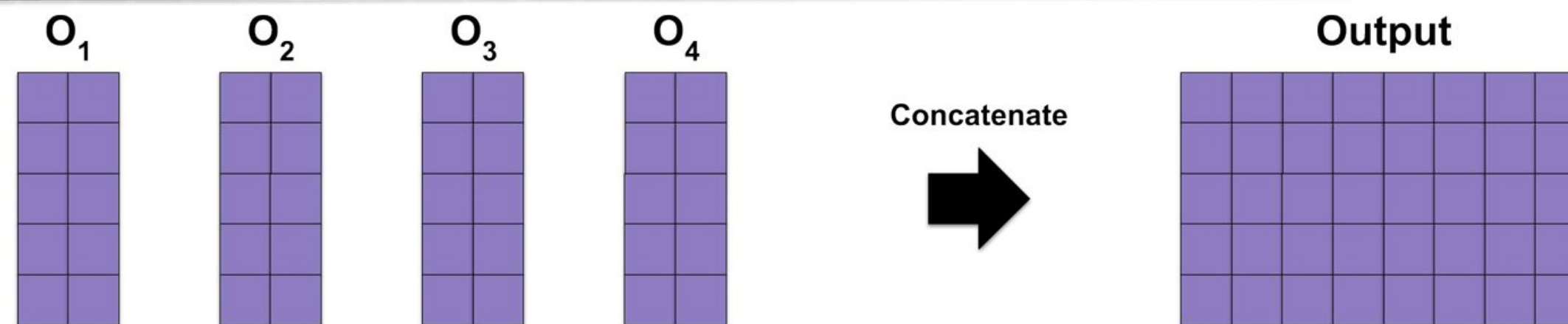
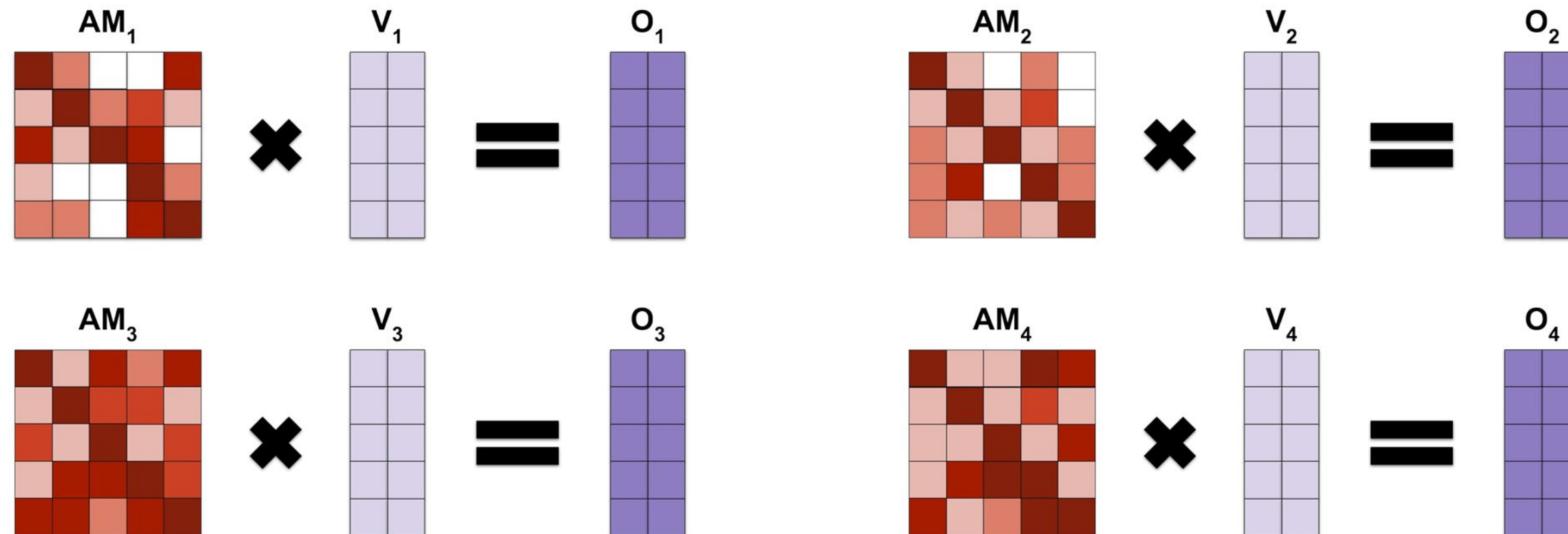
Multi-head Attention



Multi-head Attention



Multi-head Attention

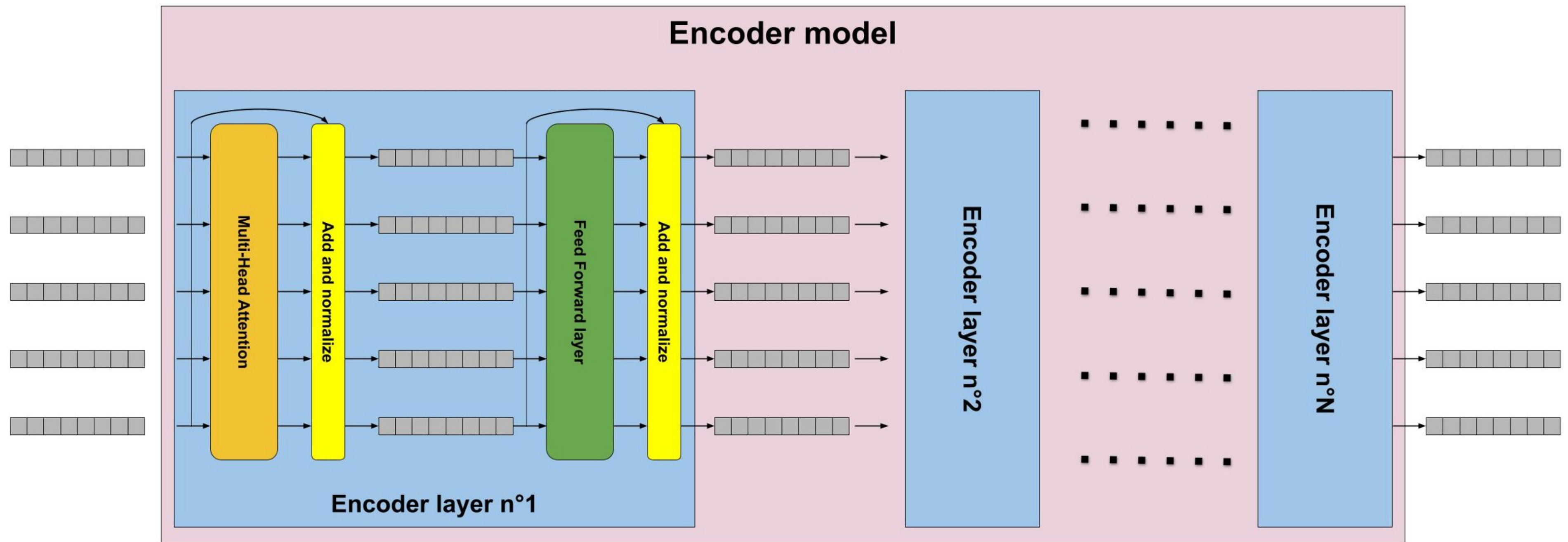


Transformer architectures

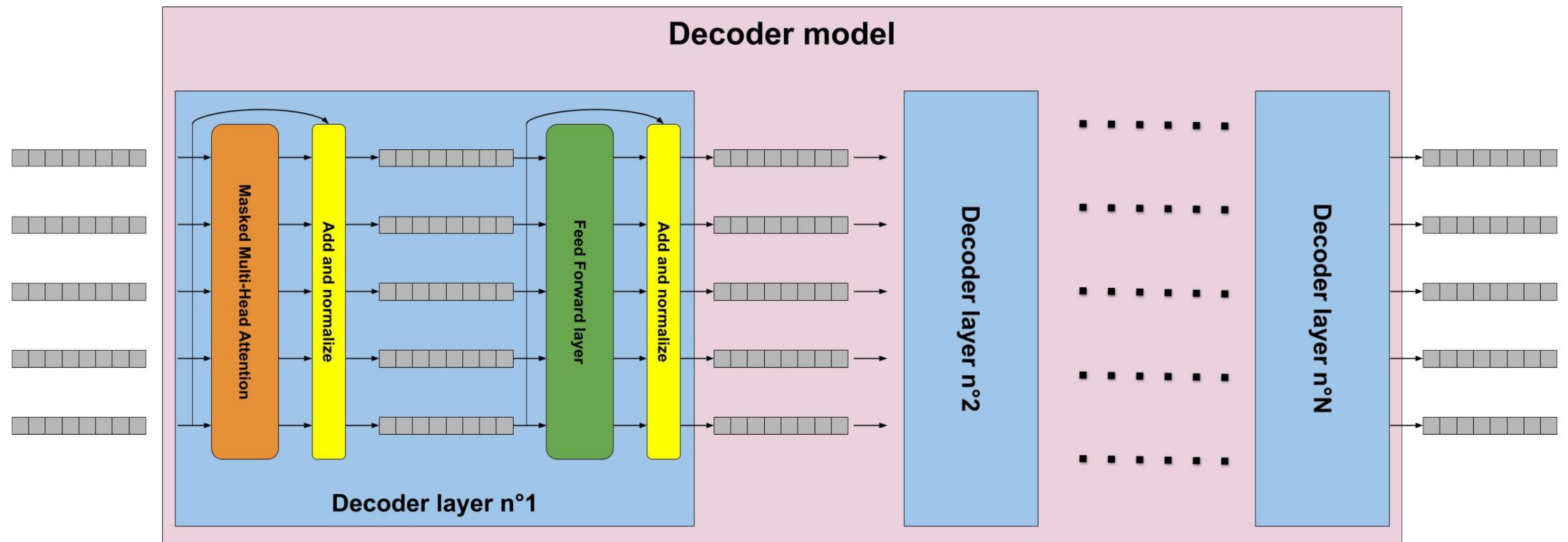
Encoder-Decoder



BERT / Encoder / Auto-encoding

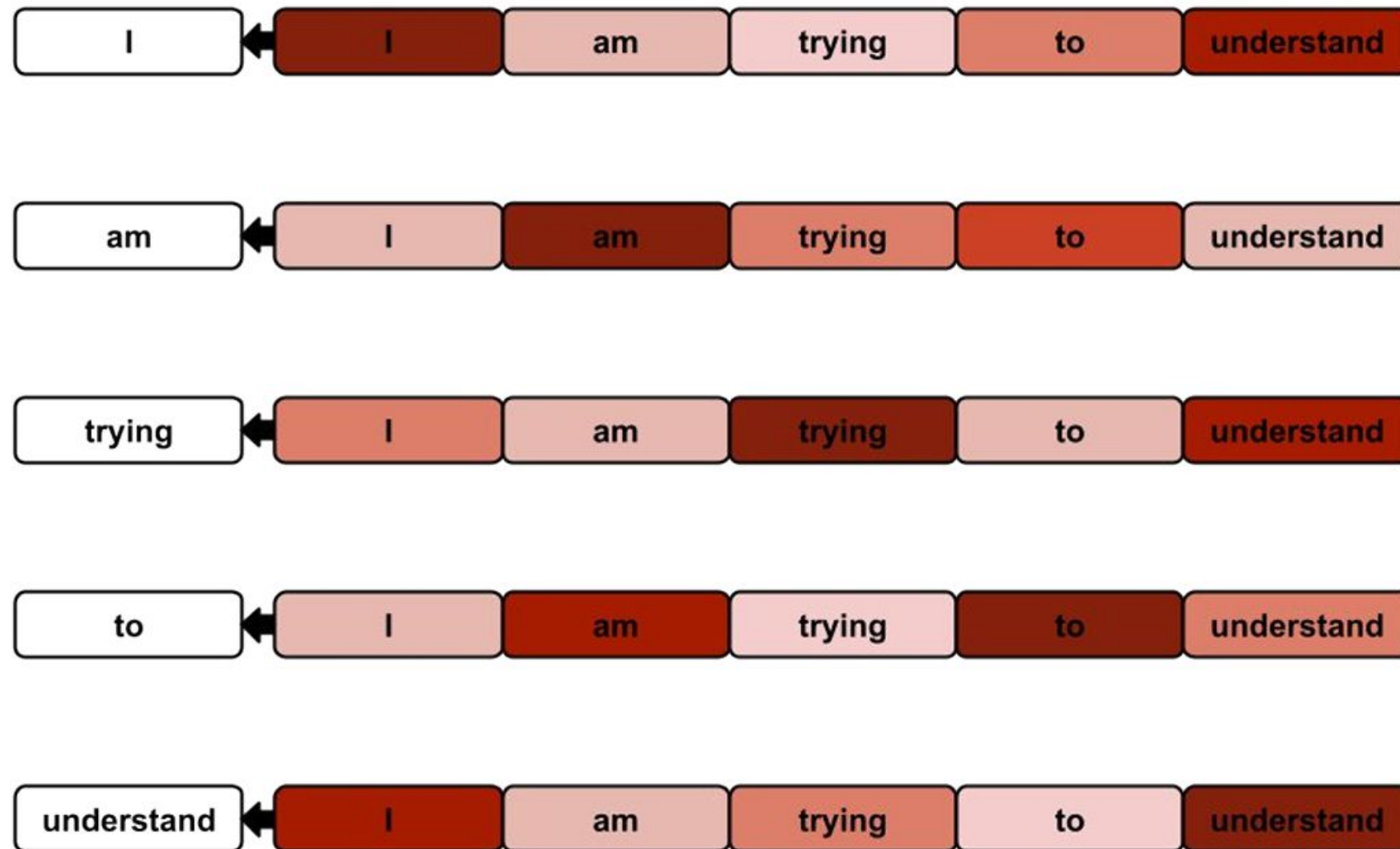


GPT / Decoder / Auto-regressive

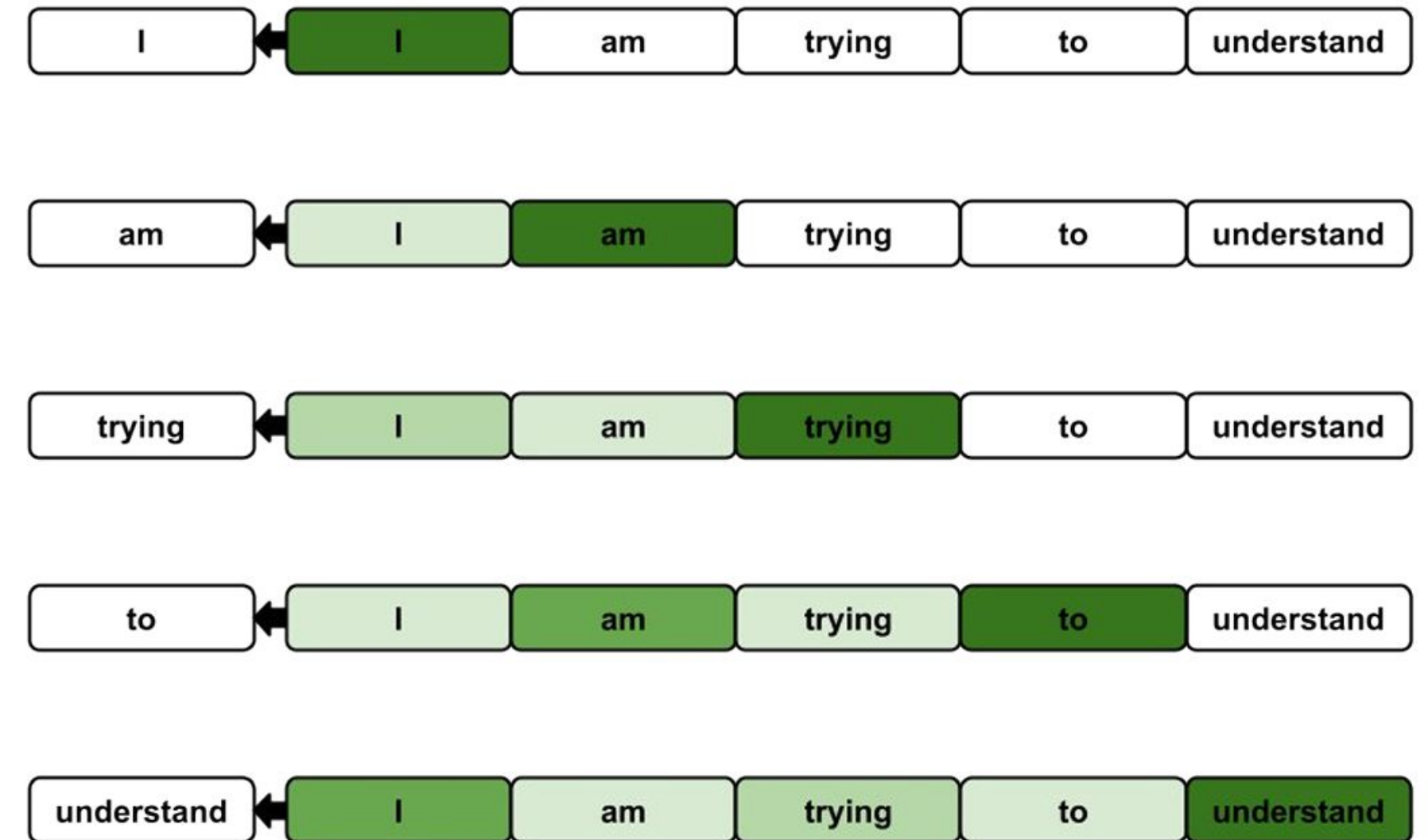


Bidirectional vs unidirectional attention

Bidirectional attention for Encoder

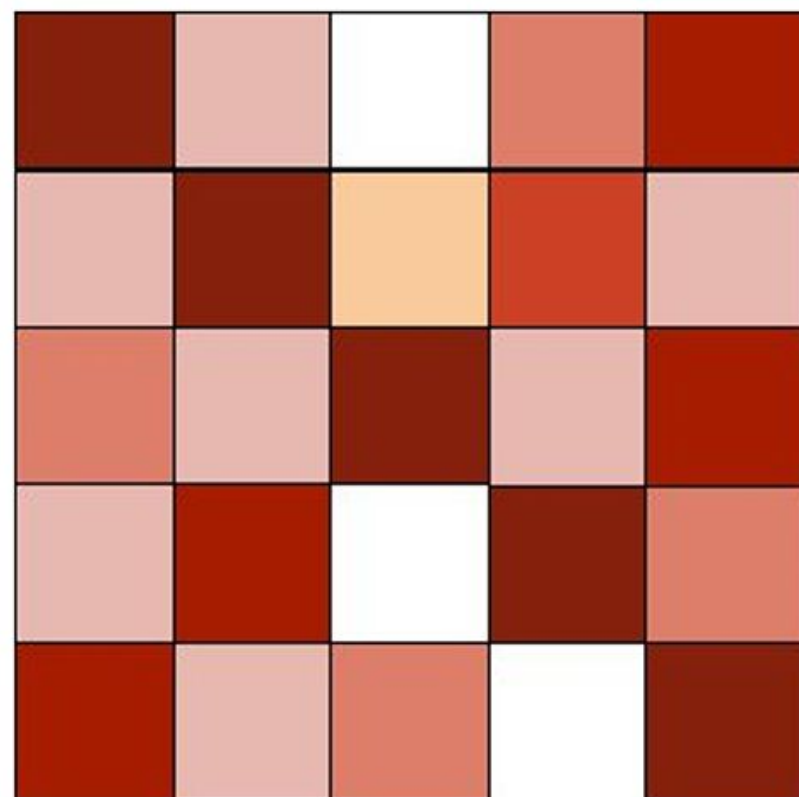


Unidirectional attention for Decoder

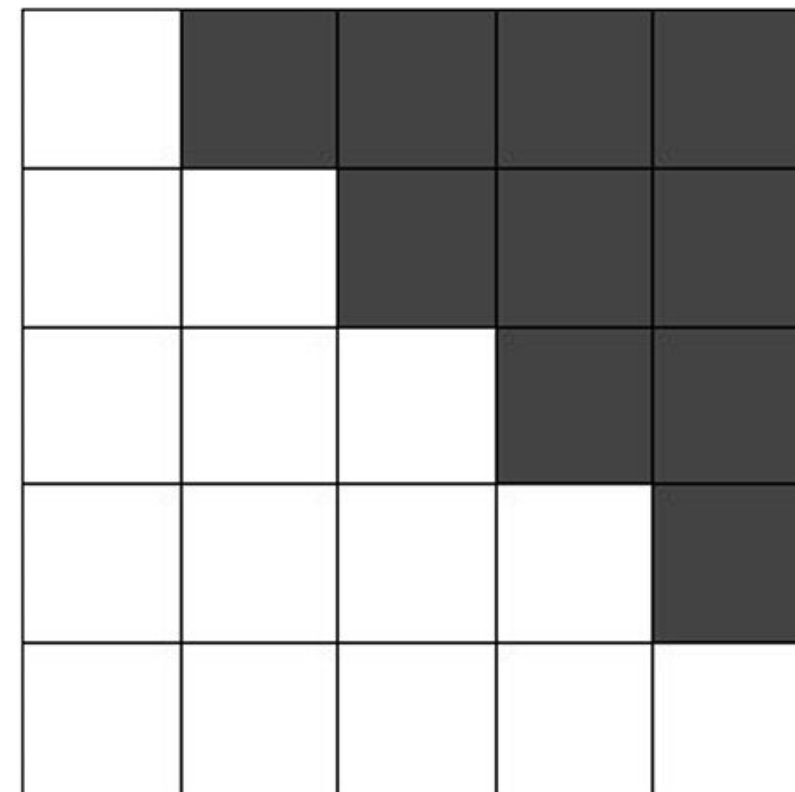


VS

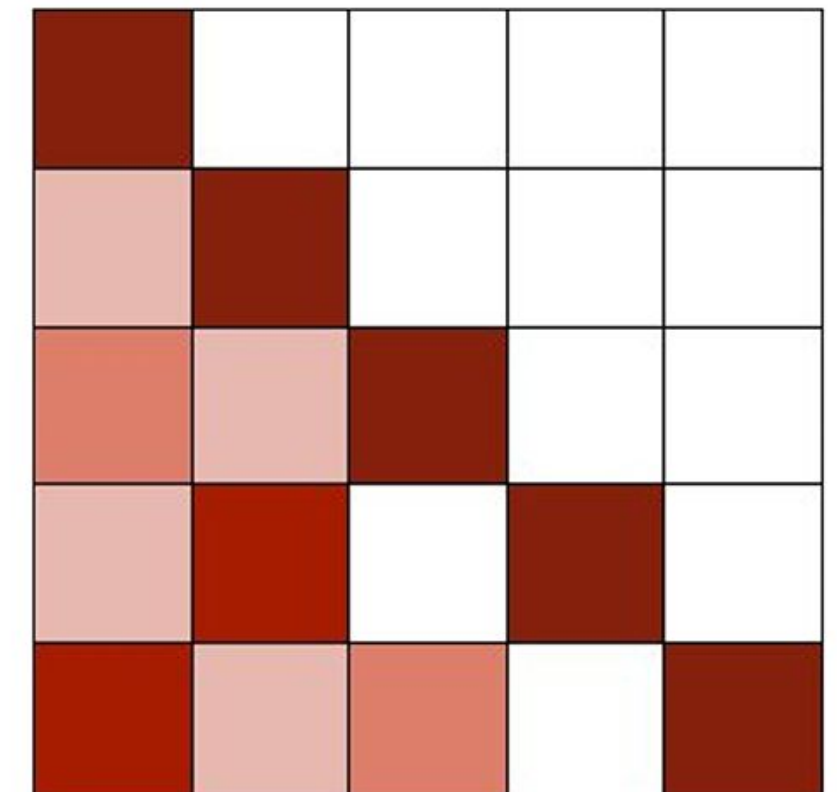
Mask attention



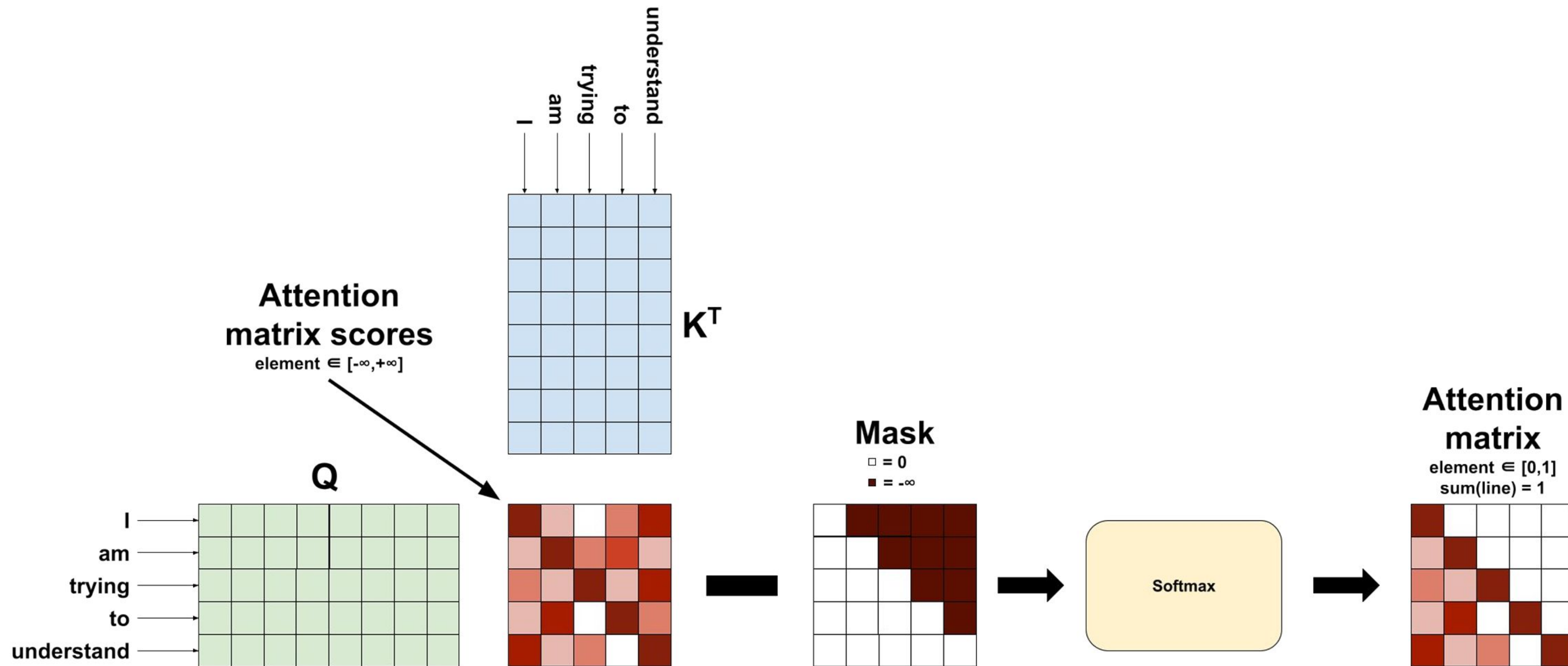
−



=

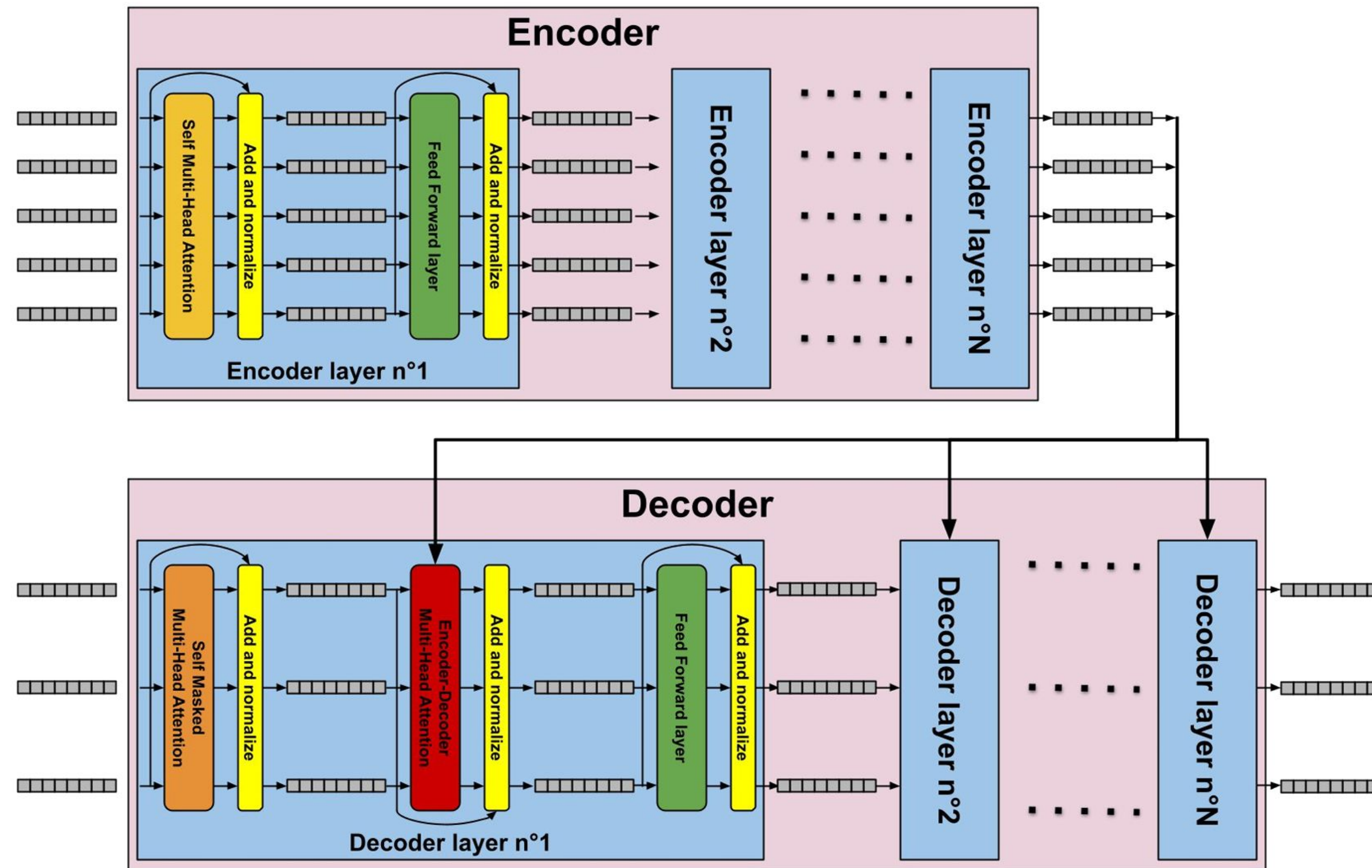


Unilateral attention detailed

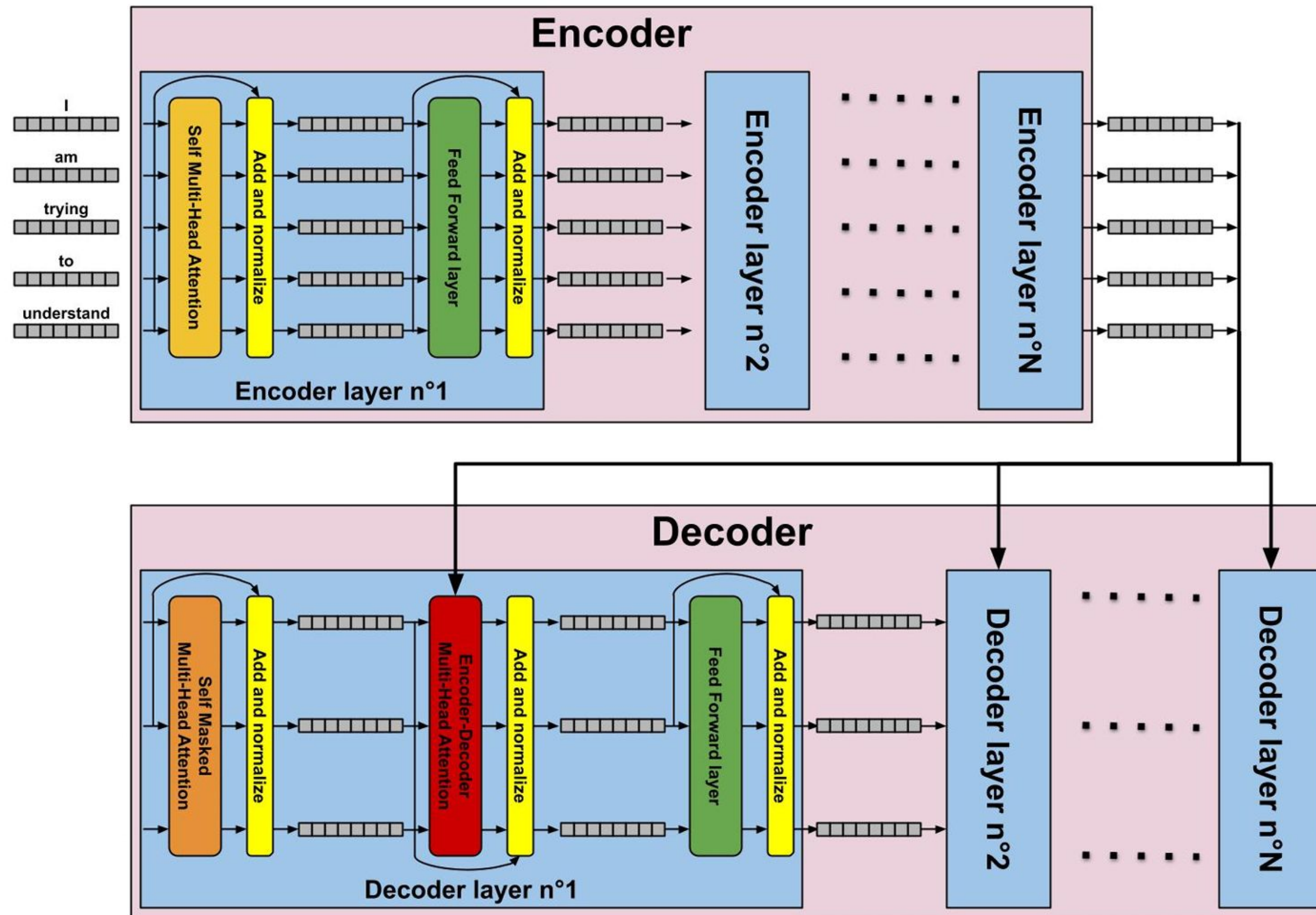


Encoder-decoder architecture

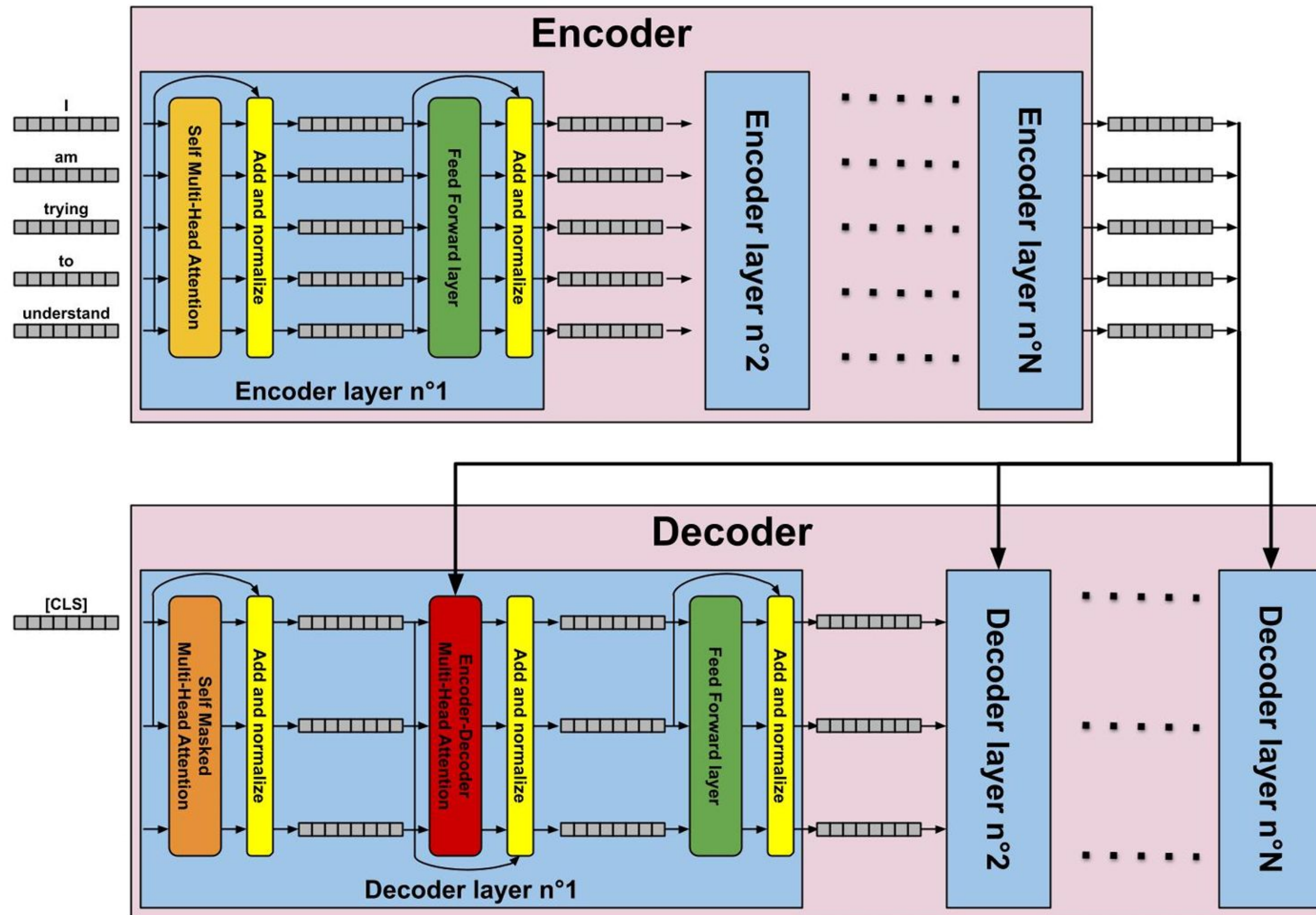
T5 / Encoder-Decoder / Sequences-to-Sequences



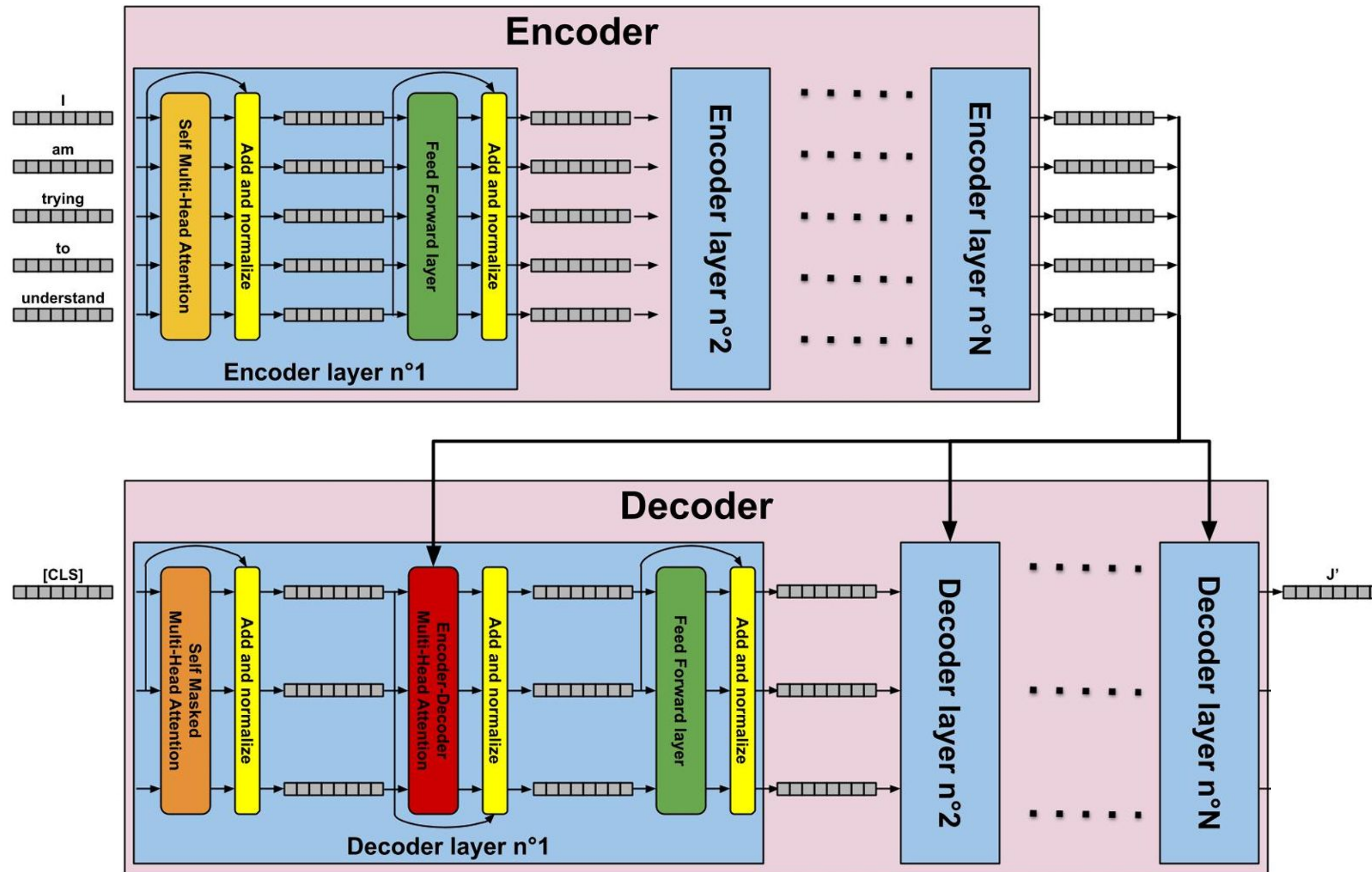
Encoder-decoder translation example



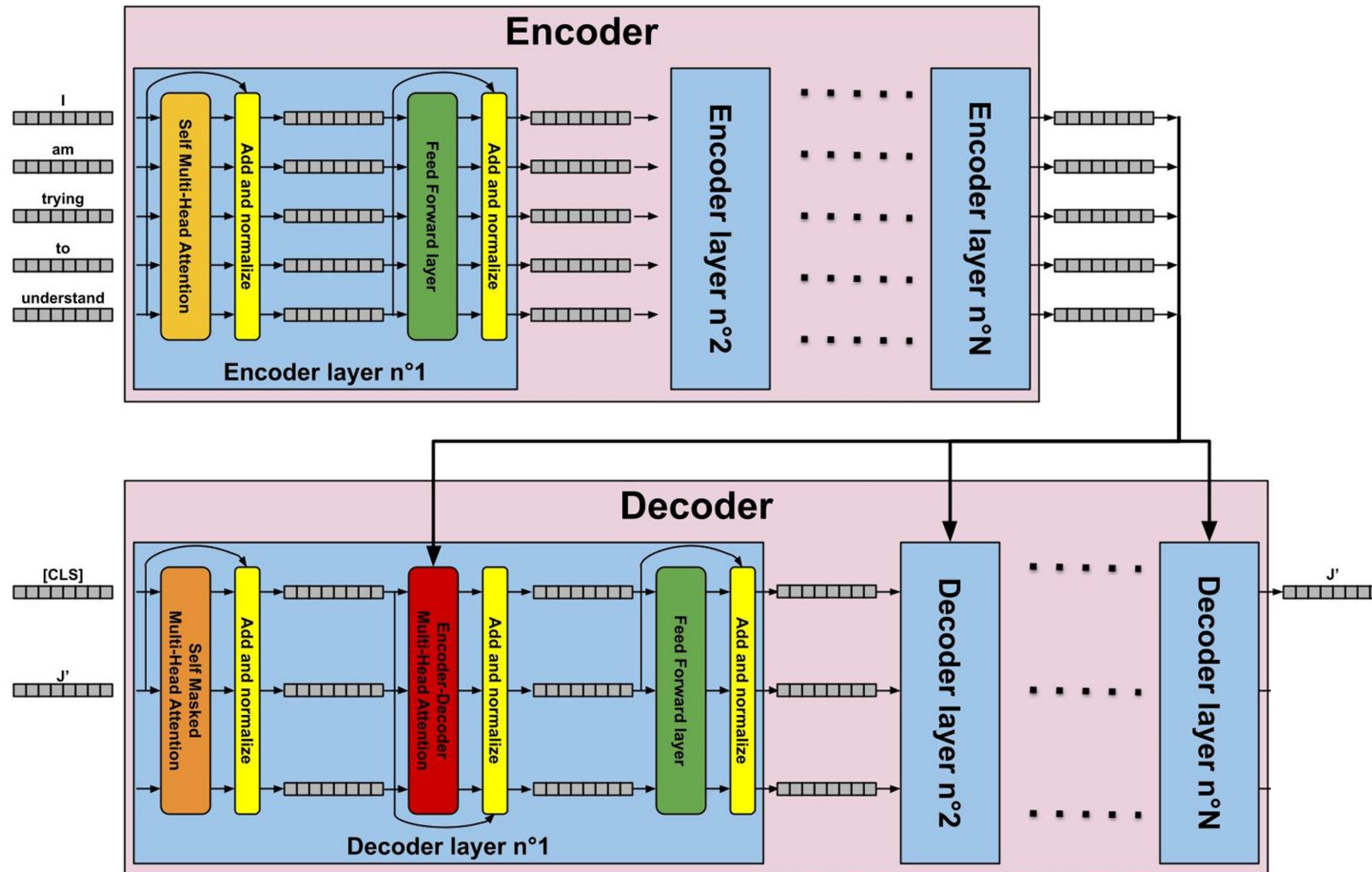
Encoder-decoder translation example



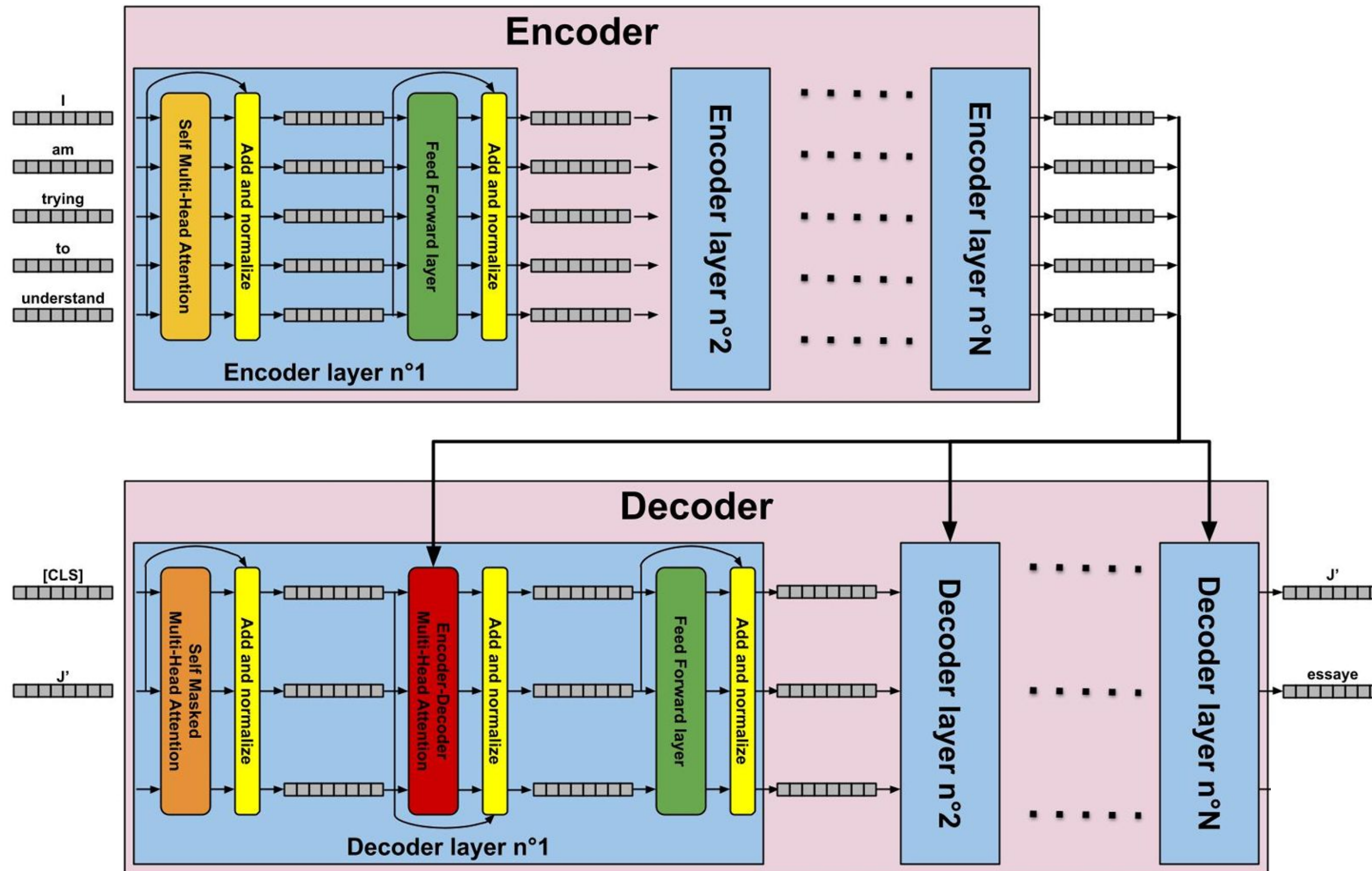
Encoder-decoder translation example



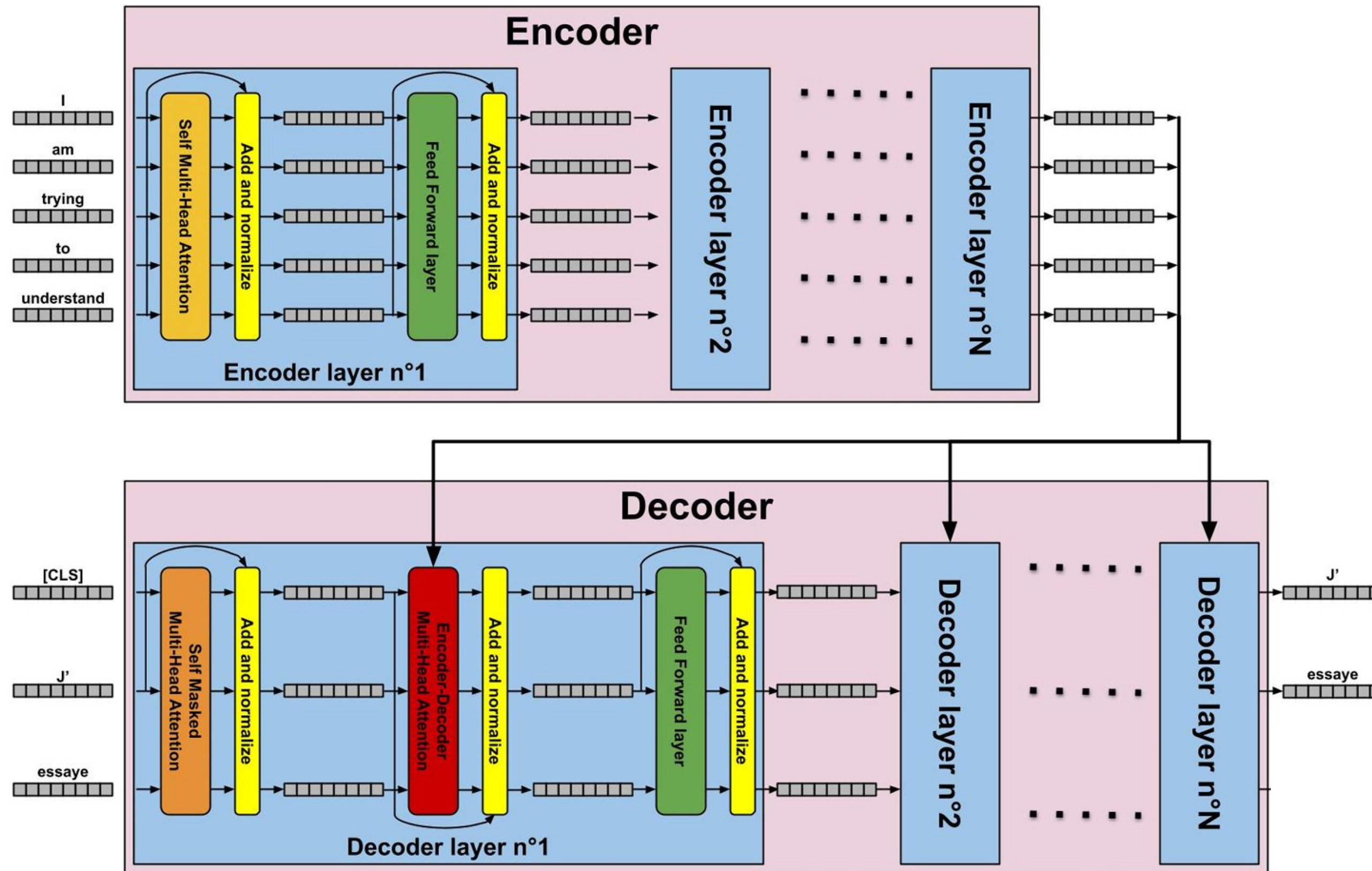
Encoder-decoder translation example



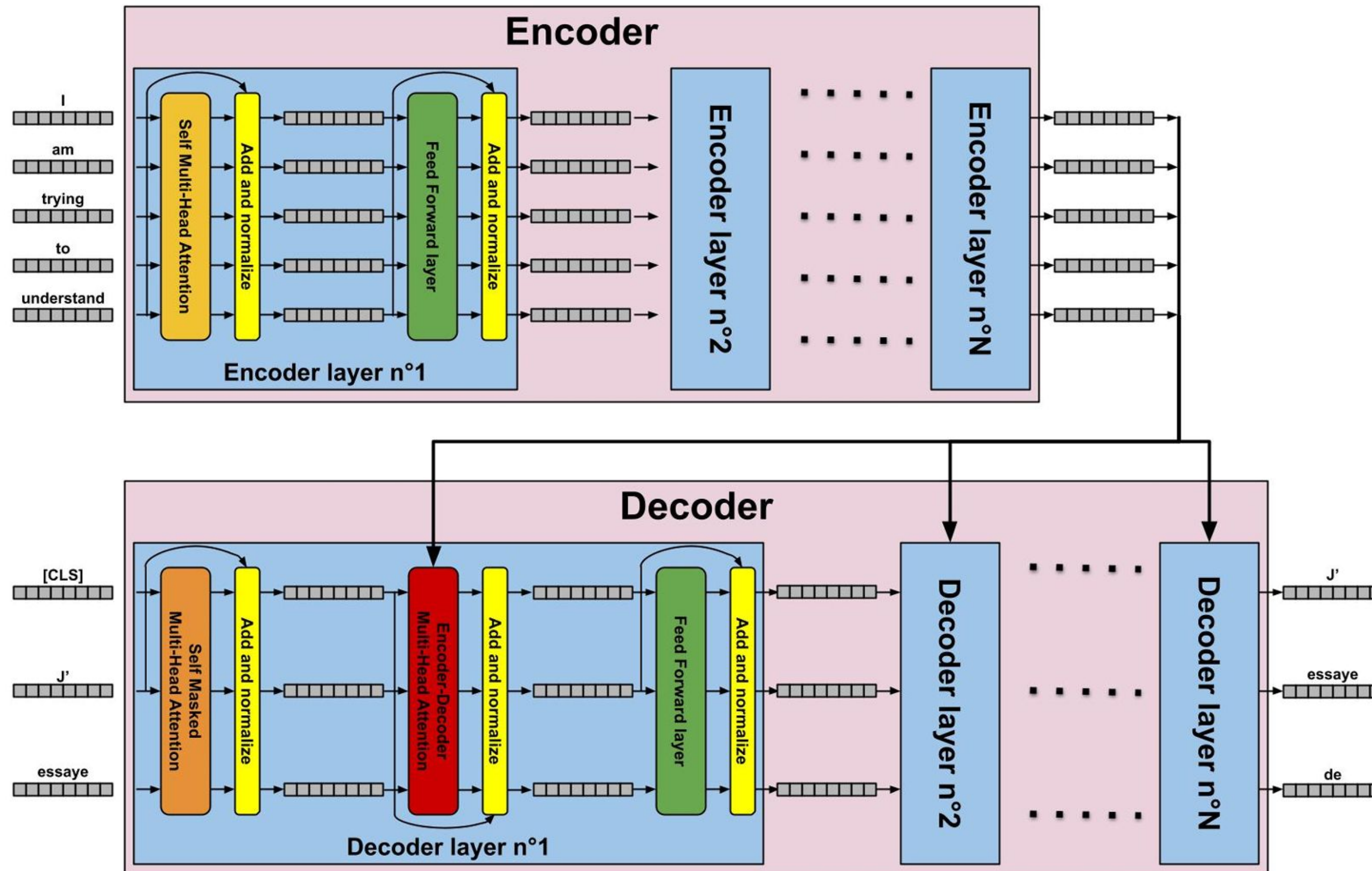
Encoder-decoder translation example



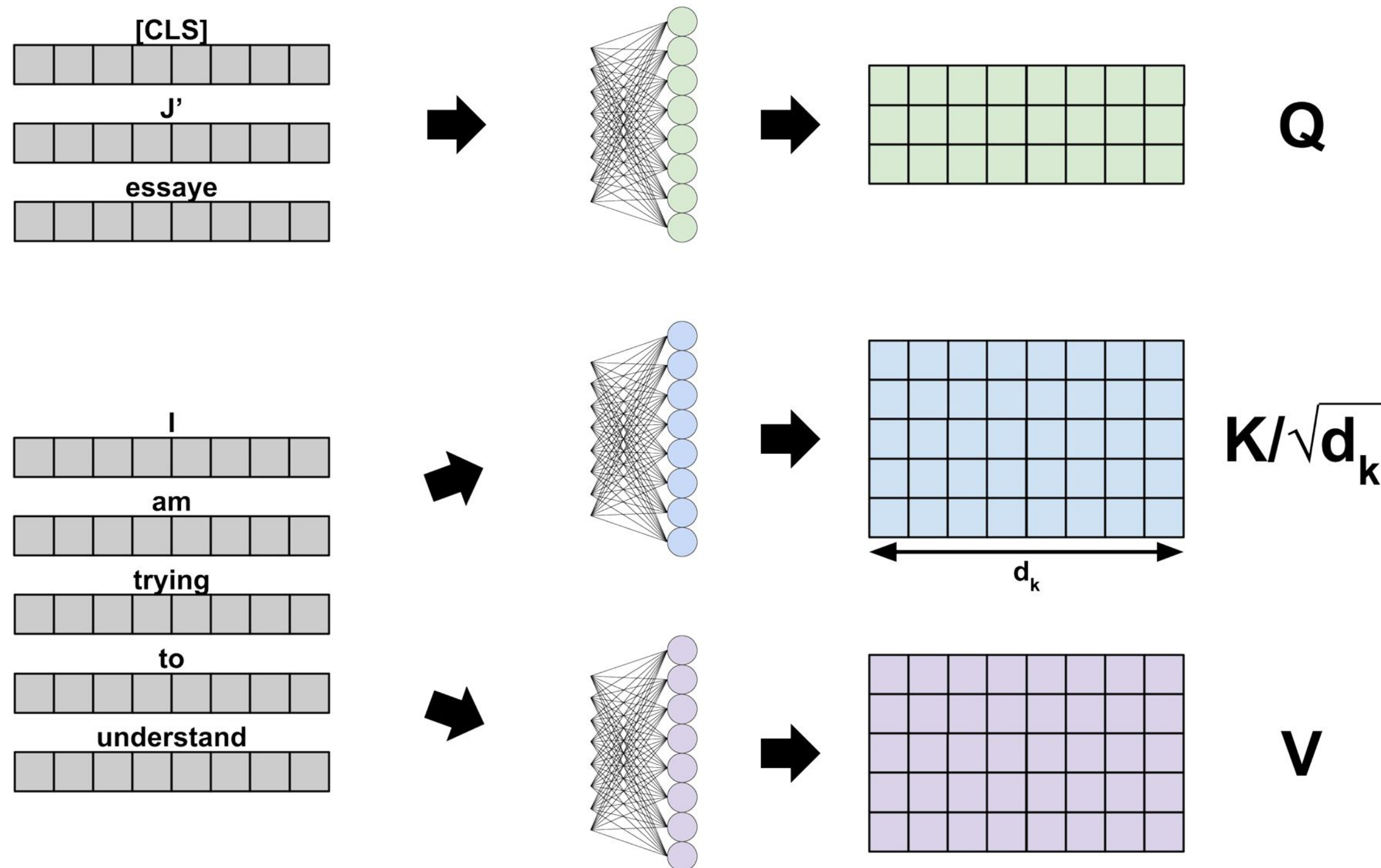
Encoder-decoder translation example



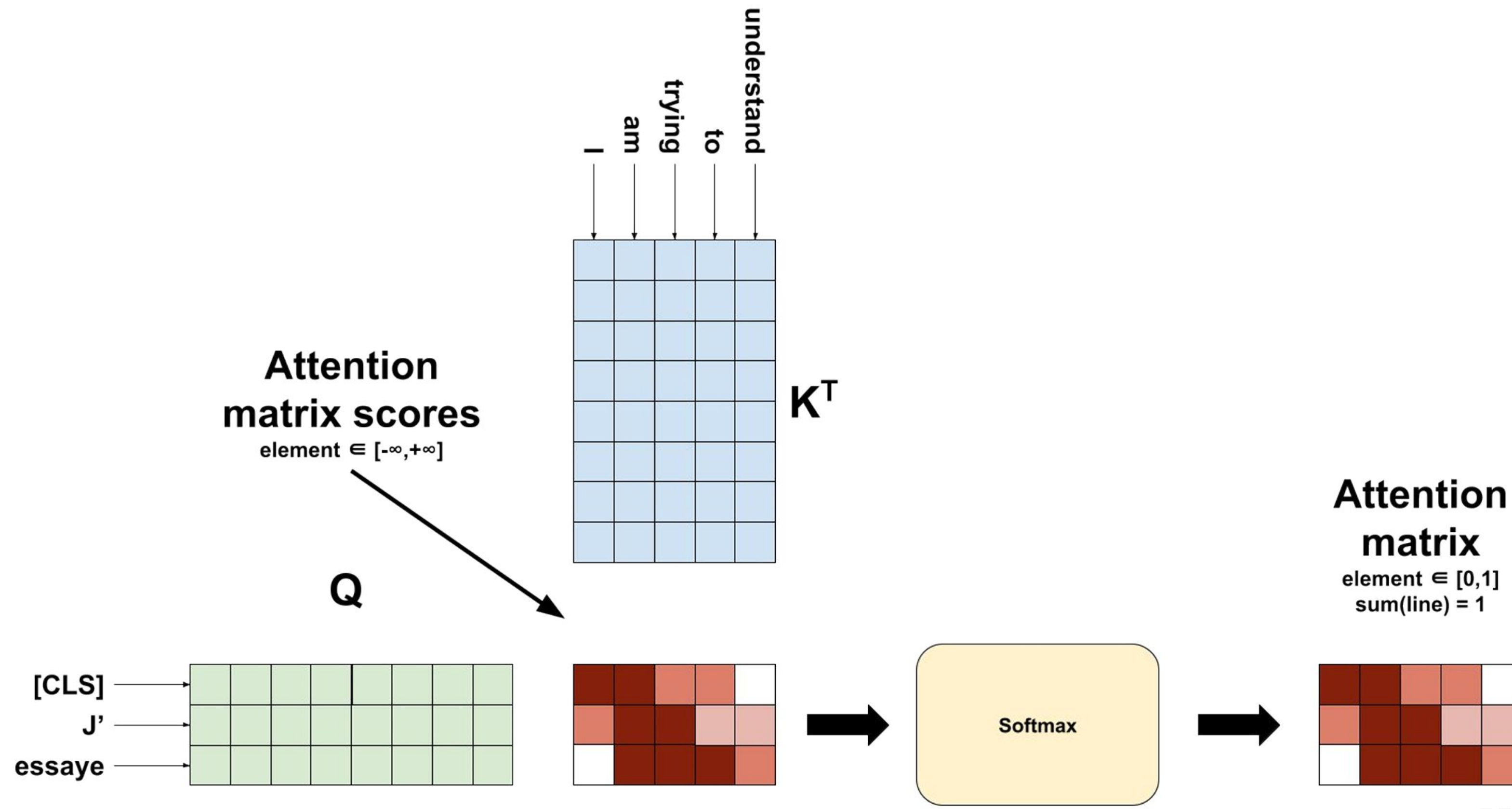
Encoder-decoder translation example



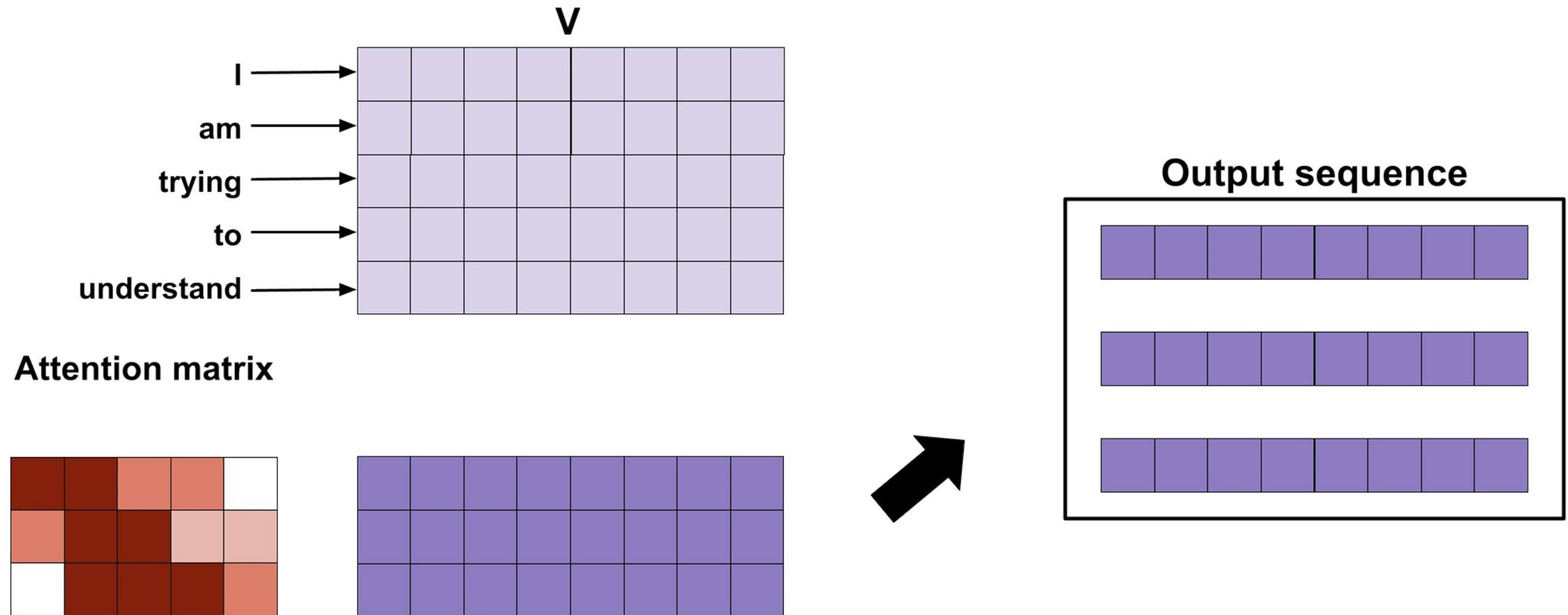
Encoder-decoder attention



Encoder-decoder attention



Encoder-decoder attention



Training a language model

I want to make a chatbot to help people better understand the civil code.



I want to make a bot which can filter out respectful comments for a real reddit experience.



Training a language model

What do you need to train a large language model ?

A truckload of data

Transformers are ravenous. You need to feed them with a substantial and hard-to-come-by dataset.



A mighty compute infrastructure

GPUs with high throughput and large VRAM can execute this training in a reasonable amount of time.



A copious amount of electricity

Storage, memory and compute power consume a lot of electricity.

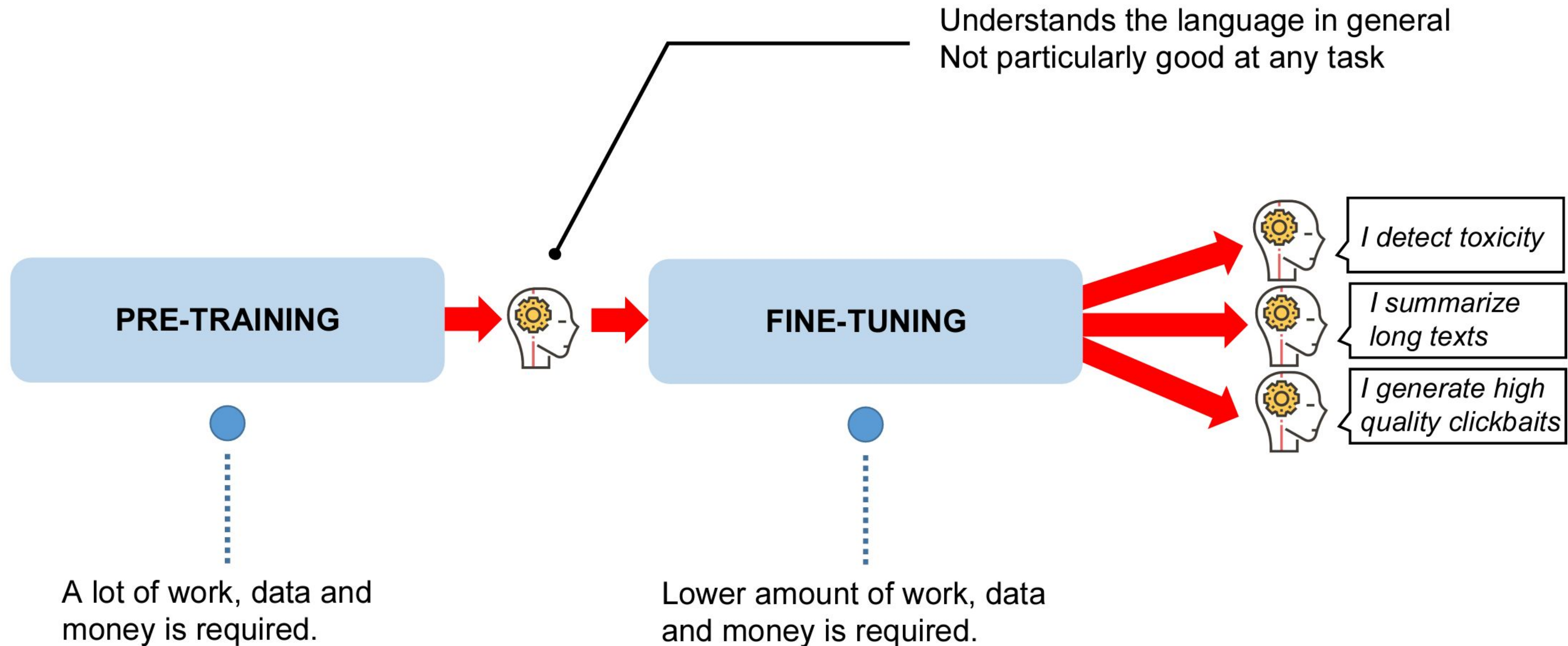


An abundance of manpower

Cleaning the dataset, making experiments, monitoring SOTA advancements is a lot of work.



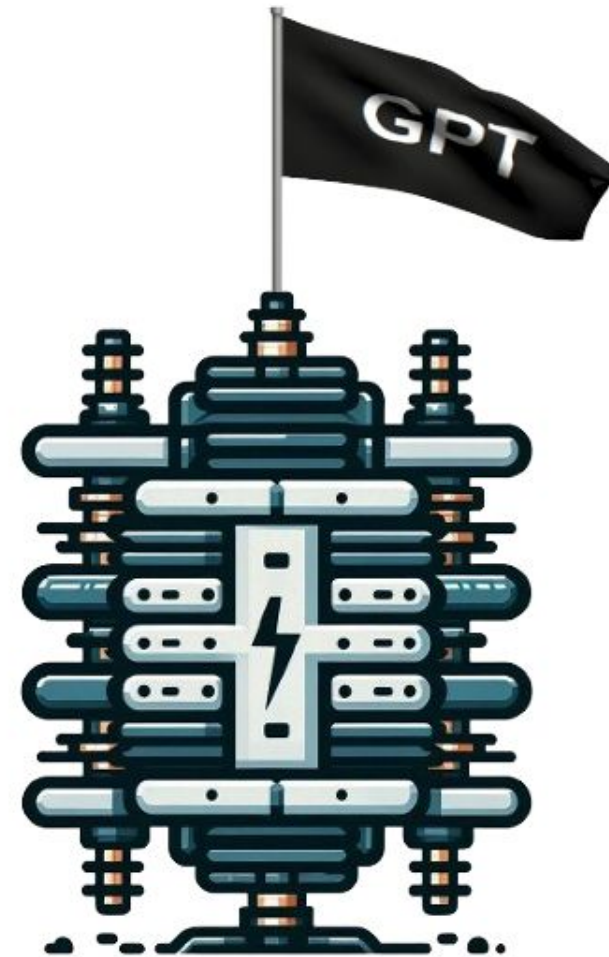
Training a language model



Pre-training a GPT-style Transformer

Input

A
transformer
is
a
deep
learning
model
that
adopts
the
mechanism
of
self

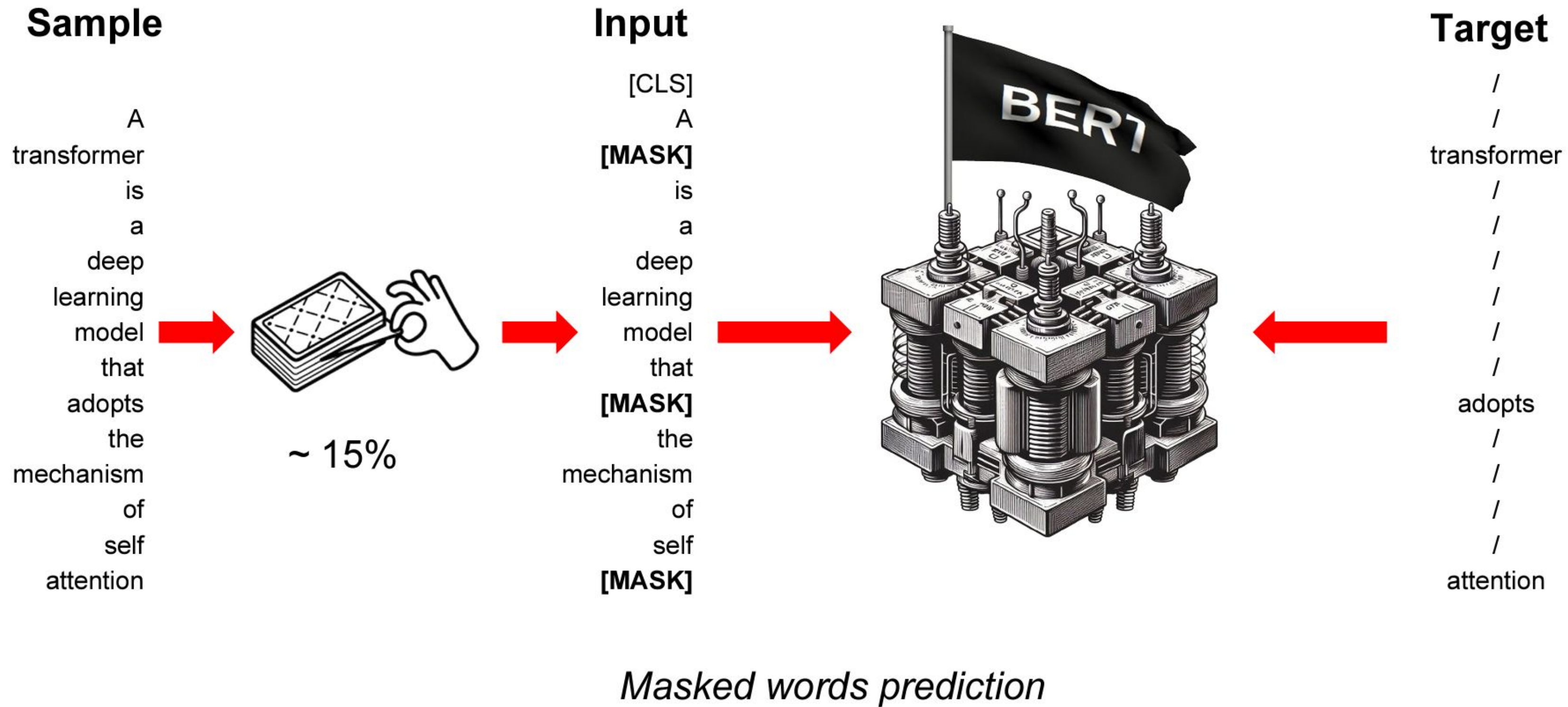


Target

transformer
is
a
deep
learning
model
that
adopts
the
mechanism
of
self
attention

Next word prediction

Pre-training a BERT-style Transformer



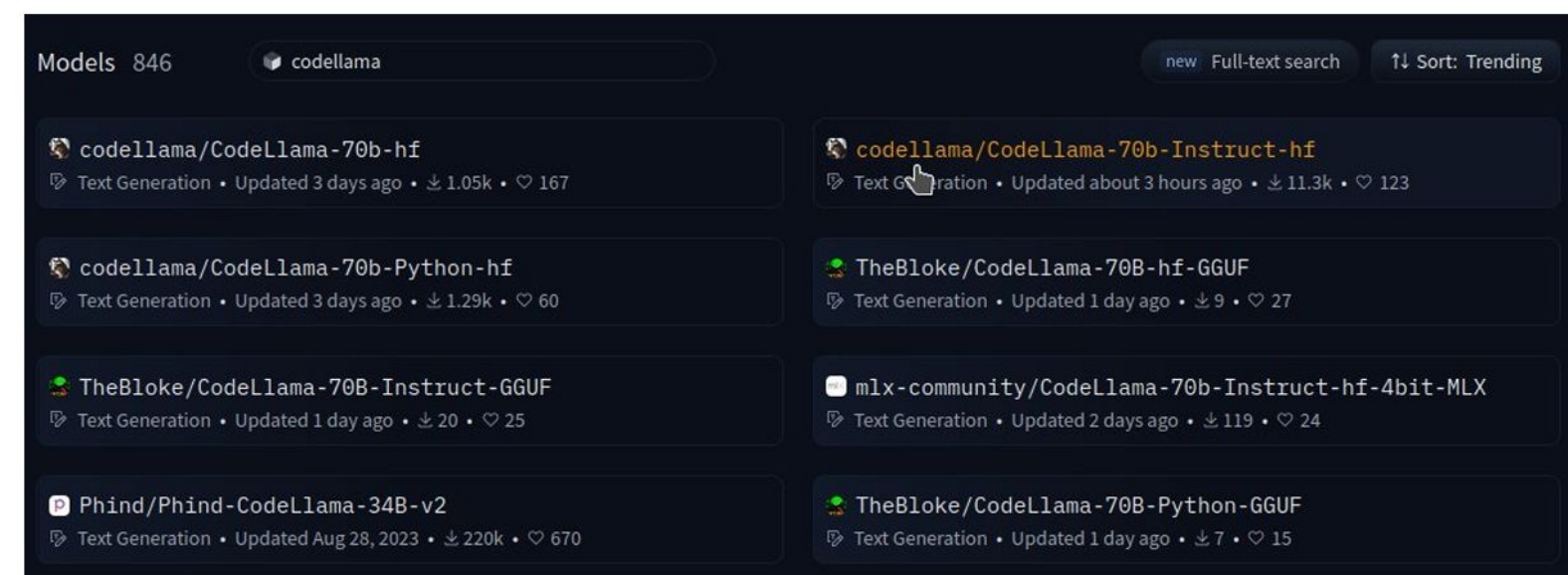
Finding a pre-trained model

The largest free pretrained transformer models database

1 — Go to <https://huggingface.co/>



2 — Look for a model



3 — Download the model

```
ncassereau@jean-zay:/fidle/transformers$ git clone https://huggingface.co/codellama/CodeLlama-70b-Instruct-hf
```

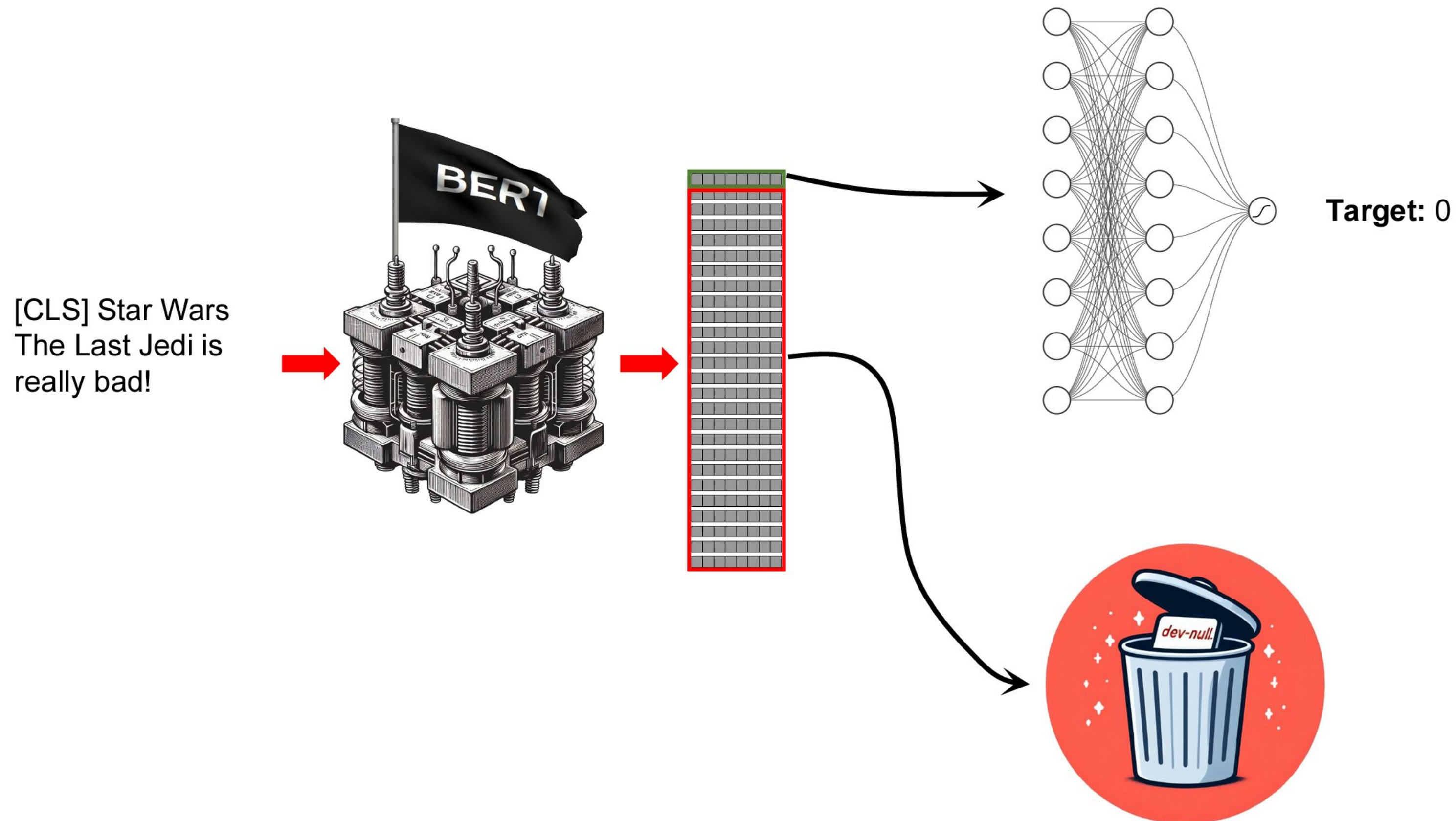
4 — Enjoy

```
1 from transformers import AutoTokenizer, AutoModelForCausalLM
2
3 path = "/fidle/transformers/CodeLlama-70b-Instruct-hf"
4 tokenizer = AutoTokenizer.from_pretrained(path)
5 model = AutoModelForCausalLM.from_pretrained(path)
6
7 input = tokenizer(["I love Jean Zay", "Hatim truly is the best"], padding=True)
8 output = model(**input)
9
```

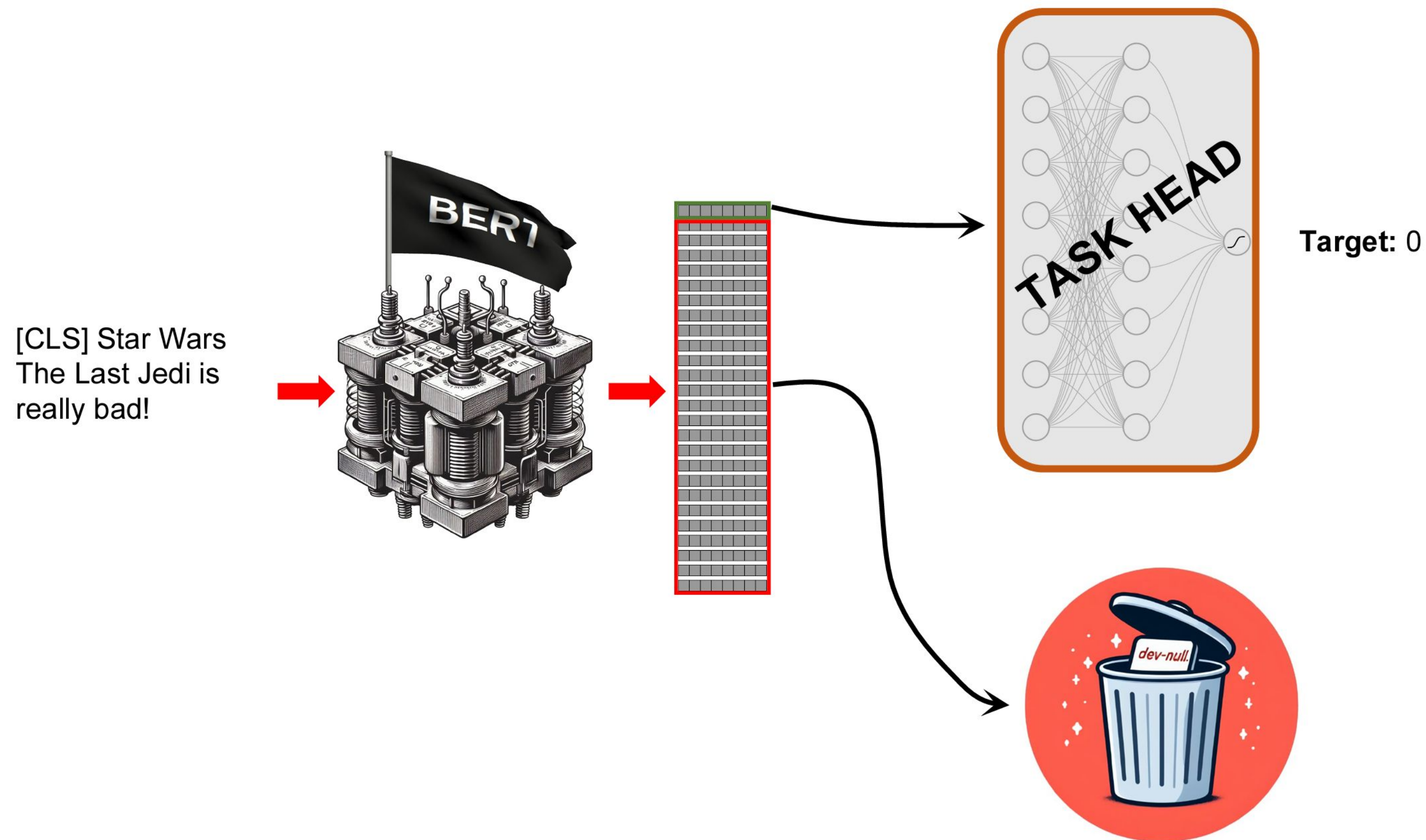

Fine-tuning of language models



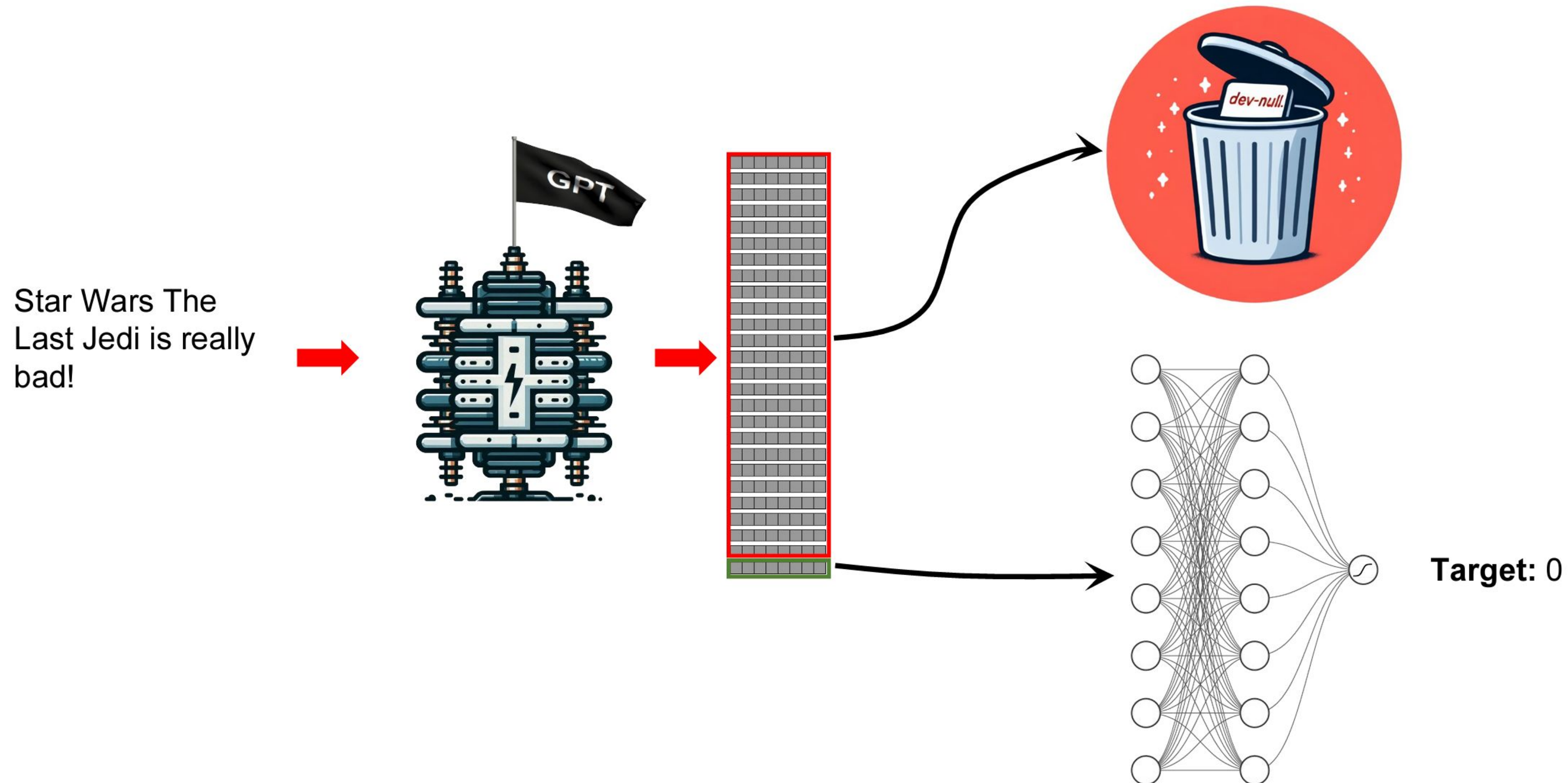
Fine-tuning a BERT - sentence classification



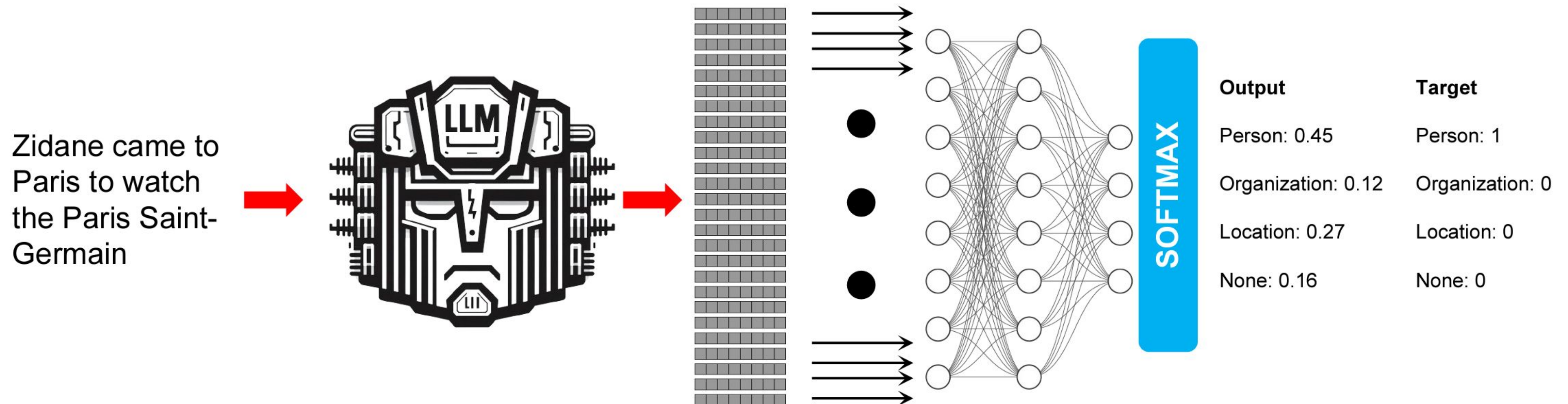
Fine-tuning a BERT - sentence classification



Fine-tuning a GPT - sentence classification



Fine-tuning a Transformer - named entity recognition



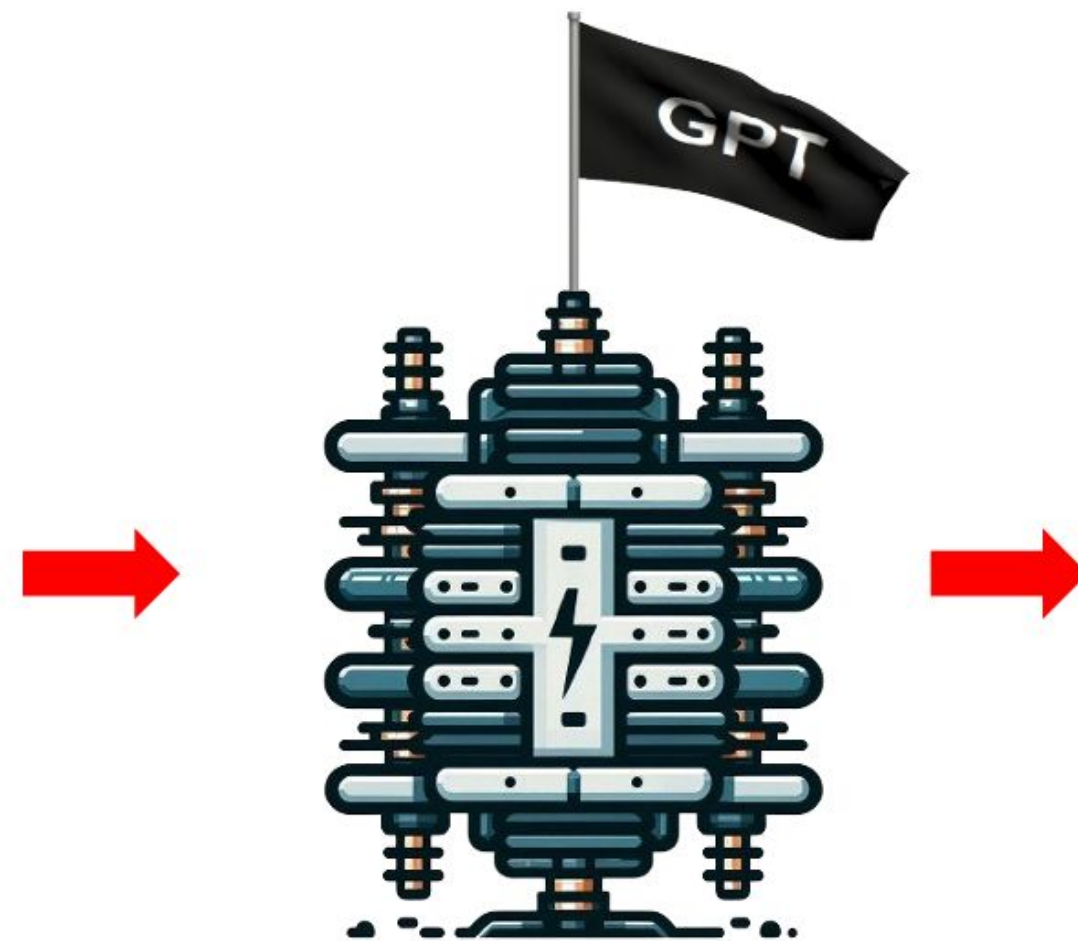
Fine-tuning a GPT with prompting

Review

This film is really trash!

Template

{{ REVIEW }} This
review is (positive,
negative or neutral):



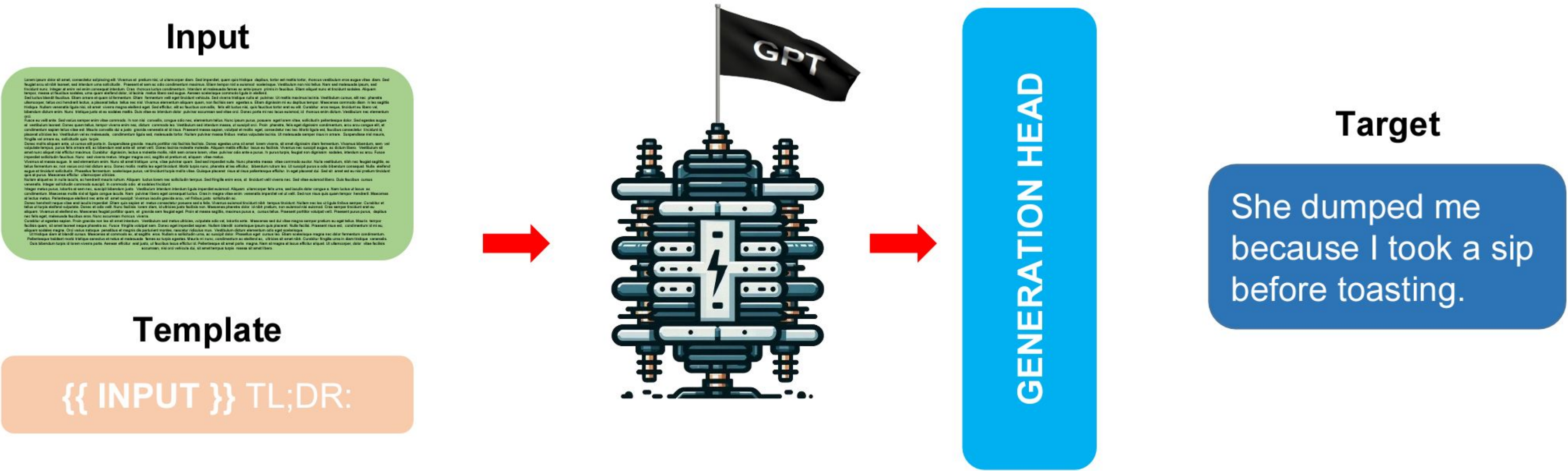
Output

Positive: 0.18
Negative: 0.44
Neutral: 0.38

Target

Positive: 0
Negative: 1
Neutral: 0

Fine-tuning a GPT - example of summarization

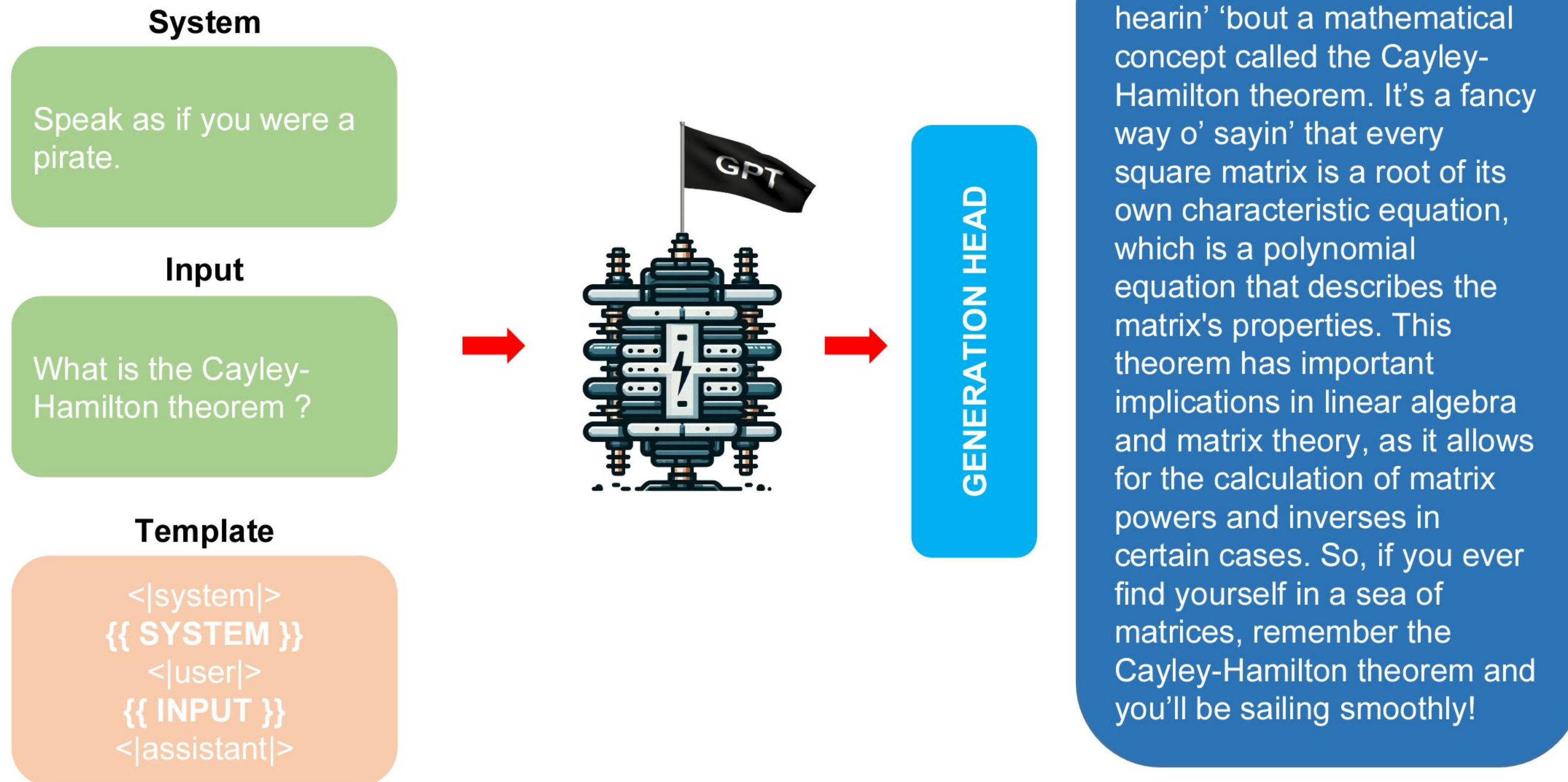


Fine-tuning a GPT with templates

Type	Task	Input ([X])	Template	Answer ([Z])
Text CLS	Sentiment	I love this movie.	[X] The movie is [Z].	great fantastic ...
	Topics	He prompted the LM.	[X] The text is about [Z].	sports science ...
	Intention	What is taxi fare to Denver?	[X] The question is about [Z].	quantity city ...
Text-span CLS	Aspect Sentiment	Poor service but good food.	[X] What about service? [Z].	Bad Terrible ...
Text-pair CLS	NLI	[X1]: An old man with ... [X2]: A man walks ...	[X1]? [Z], [X2]	Yes No ...
Tagging	NER	[X1]: Mike went to Paris. [X2]: Paris	[X1] [X2] is a [Z] entity.	organization location ...
Text Generation	Summarization	Las Vegas police ...	[X] TL;DR: [Z]	The victim ... A woman
	Translation	Je vous aime.	French: [X] English: [Z]	I love you. I fancy you. ...

Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing, Pengfei Liu and al

Orienting the style with templates



A word on ChatGPT

Transformers and ChatGPT

Genesis of ChatGPT:

2018: GPT (Improving Language Understanding by Generative Pre-Training)

→ Use of the decoder part of a pre-trained and fine-tuned transformer to perform various tasks (117 million parameters).

2019: GPT-2 (Language Models are Unsupervised Multitask Learners)

→ Q/A with a natural language prompt and human-like responses (1.5 billion parameters).

2020: GPT-3 (Language Models are Few-Shot Learners)

→ Improvement of the previous model with a much larger network (175 billion parameters, 96 layers).

2022: ChatGPT & InstructGPT (Training language models to follow instructions with human feedback)

→ Model enhancement using supervised learning and reinforcement learning.

2023: GPT-4

→ Much larger model than its predecessor, multimodal (accepts both images and text as input) (1.8 trillion parameters).

A word on ChatGPT

Transformers and ChatGPT



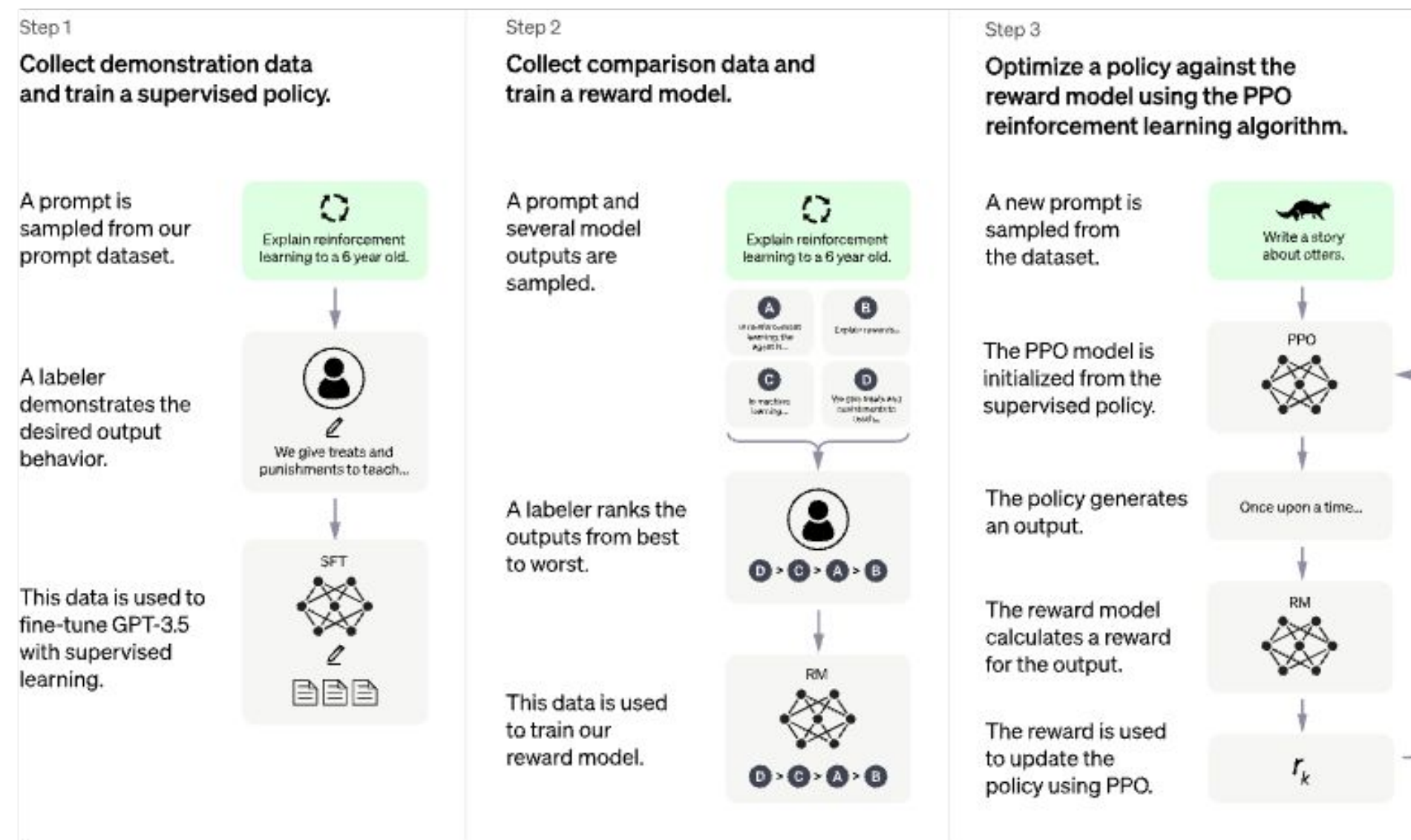
Characteristics (February 2025)

- Based on the GPT-4o mini architecture (GPT-4 and GPT-4o for the paid version).
- Training data collected up until October 2023.
- Available in multiple languages (English, French, Spanish, Chinese, etc.).
- Supports text, image, video, and audio as input.
- Output: text, with image, video, and audio being gradually introduced.
- Model sizes:
 - GPT-4: 1.8 trillion parameters
 - GPT-4o: 200 billion parameters
 - GPT-4o mini: 8 billion parameters

Learning of ChatGPT

In 4 Main Steps

- Unsupervised learning of the underlying GPT-4 model.
- Supervised fine-tuning of ChatGPT on human-written question/answer examples.
- Supervised training of a response evaluation model.
- Reinforcement learning fine-tuning, using the evaluation model to improve ChatGPT's performance.

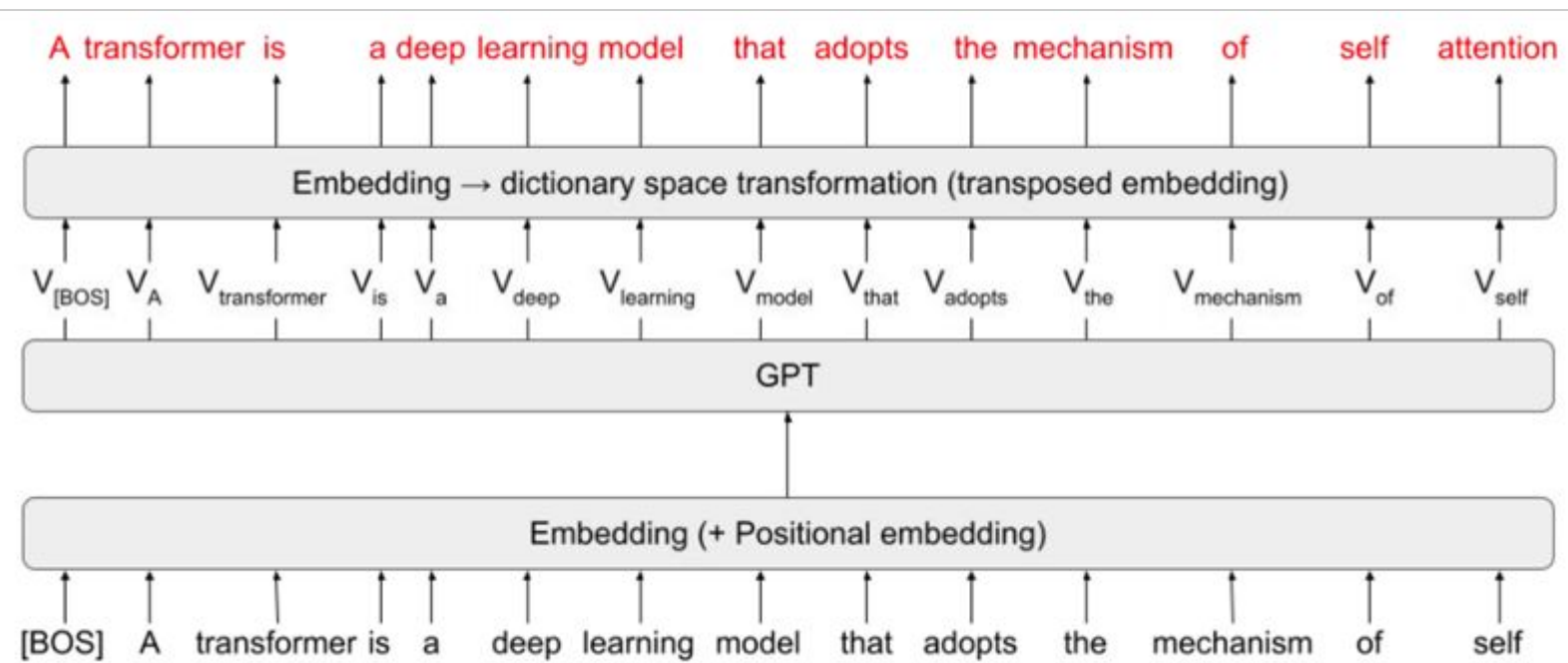


Step 0: Unsupervised Learning of the GPT-4 Model

- Objective: Learn syntactic and grammatical structures of language, as well as general knowledge across various domains.
- Data: [Typically includes vast amounts of text from books, articles, websites, and other sources.]

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

Strategy: Step-by-step prediction of the next word in a sequence of words.



Learning of ChatGPT

Step 1: Supervised Fine-Tuning of the ChatGPT Model

Objective: Adapt the model to the conversation task.

Data: Human-written dialogue sequences (question/answer examples).

Strategy: Continue training the pre-trained model using these data.

Step 2: Supervised Training of a Response Evaluation Model

Objective: Develop a model capable of evaluating the responses generated by the model—essential for Step 4.

Data: Example questions with multiple model-generated responses, ranked by humans based on quality.

Strategy: Train a neural network using this ranked data.

Step 3: Reinforcement Learning Fine-Tuning of the ChatGPT Model

Objective: Optimize the model.

Data: Example questions.

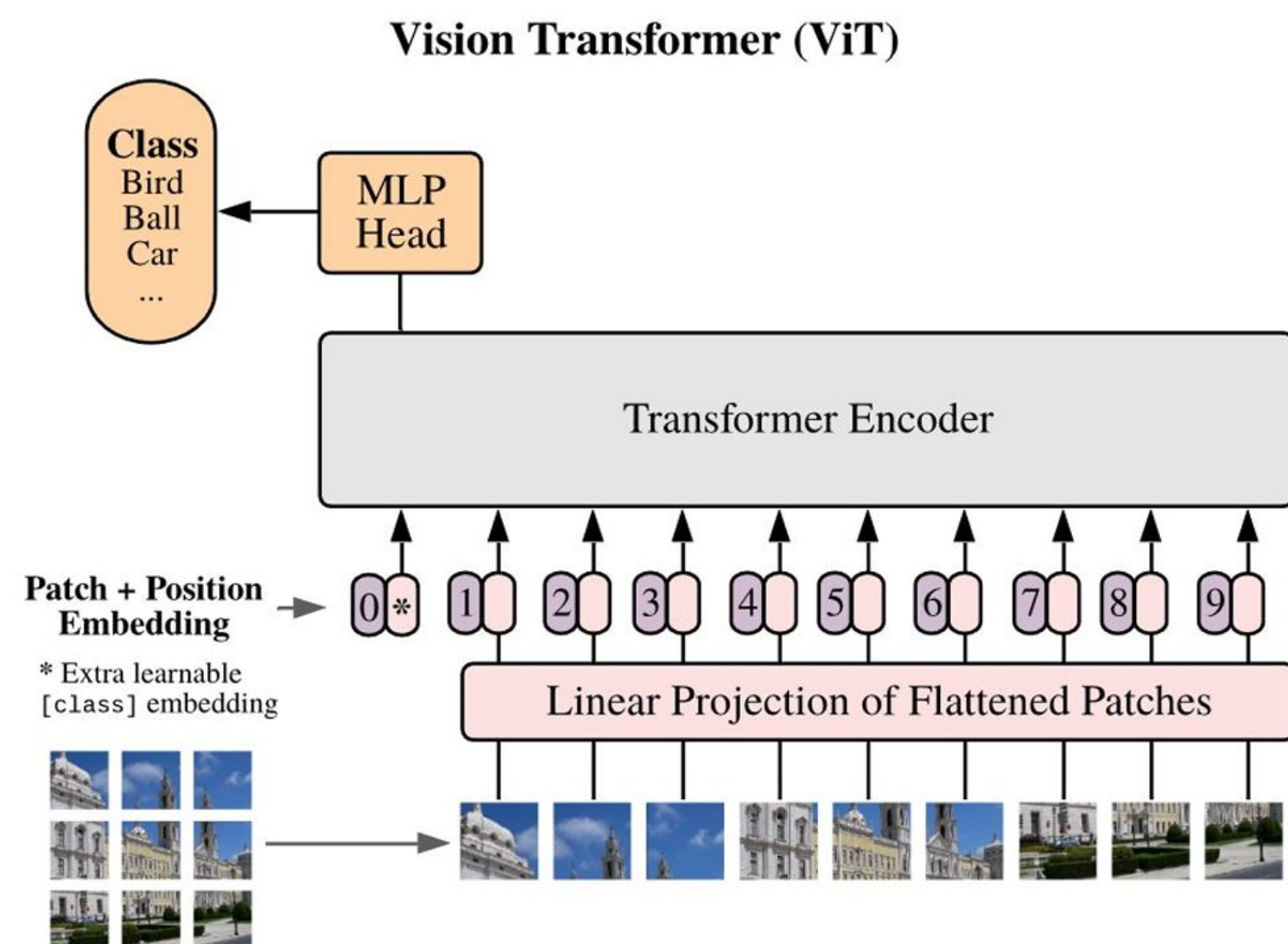
Strategy:

ChatGPT generates a response for each selected question.

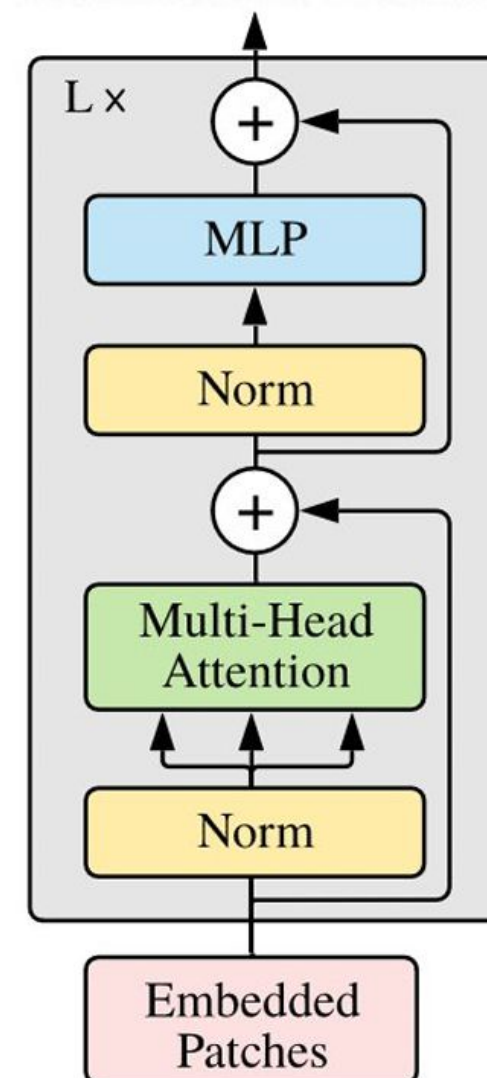
The evaluation model predicts the quality of this response.

This predicted score is used as a reward signal in reinforcement learning (using the Proximal Policy Optimization algorithm) to improve the model's response generation quality.

Vision Transformers



Transformer Encoder



Input

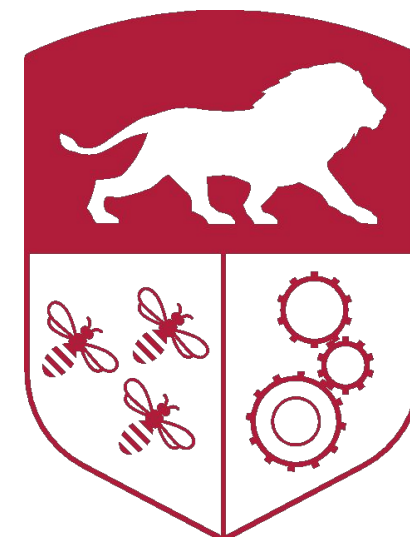


Attention



Some useful references

- Fidle - Deep Learning Introduction (<https://www.fidle.cnrs.fr/w3/>)
- CS231n: Convolutional Neural Networks for Visual Recognition (<http://cs231n.stanford.edu>)
- Neural Networks and Deep Learning (<http://neuralnetworksanddeeplearning.com>)
- Deep Learning (<http://www.deeplearningbook.org>)
- PyTorch (<http://pytorch.org>)
- Weights & Biases (<https://wandb.ai/site/>)
- Hugging Face (<https://huggingface.co/>)



**CENTRALE
LYON**

36, avenue Guy de Collongue 69130 Écully
www.ec-lyon.fr | [@centralelyon](https://twitter.com/centralelyon)