

# 机器翻译、Transformer与大模型

**林洲汉**

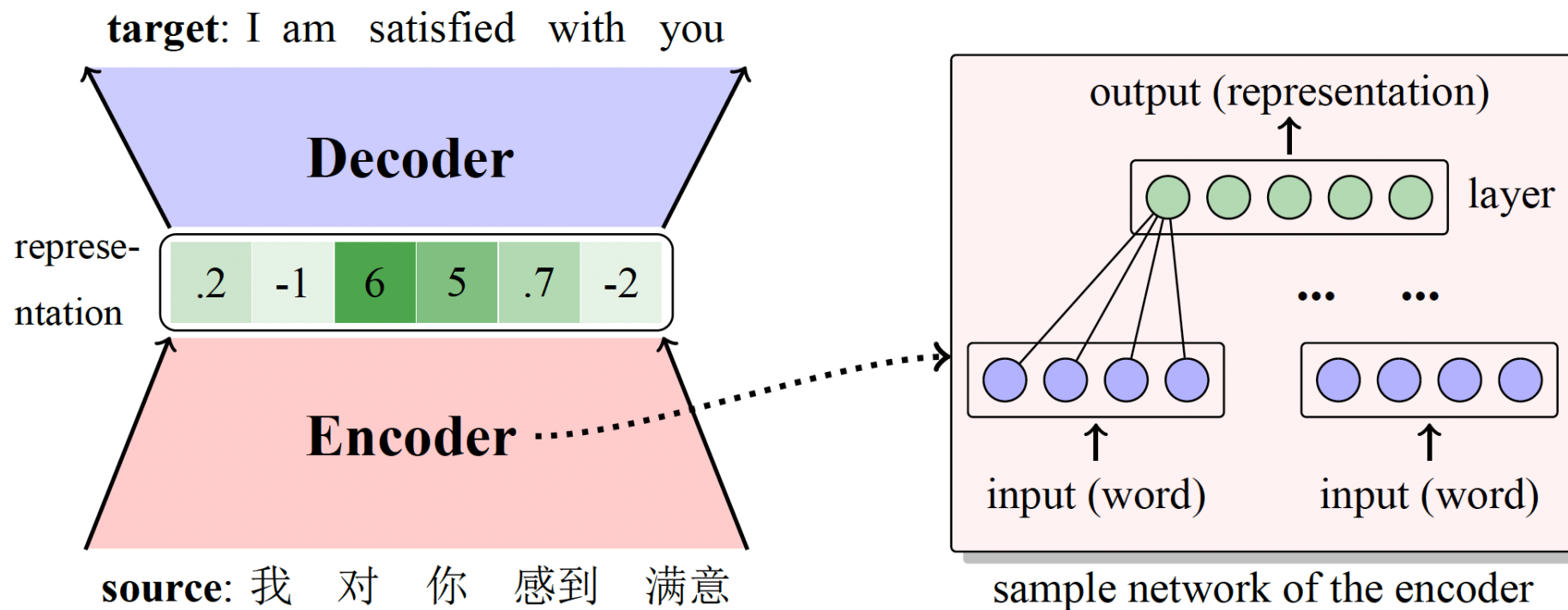
上海交大 LUMIA实验室

- ▶ **神经机器翻译**
  - ▶ 使用LSTM/GRU进行机器翻译
  - ▶ Attention机制
  - ▶ Self-attention 与Transformer模型
- ▶ **解码器 (Decoder-Only) 的大模型: GPT**
- ▶ **从GPT-3到ChatGPT: 总览**

- ▶ **神经机器翻译**
  - ▶ 使用LSTM/GRU进行机器翻译
  - ▶ Attention机制
  - ▶ Self-attention 与Transformer模型
- ▶ 解码器 (Decoder-Only) 的大模型: GPT
- ▶ 从GPT-3到ChatGPT: 总览

## ► Neural Machine Translation (NMT):

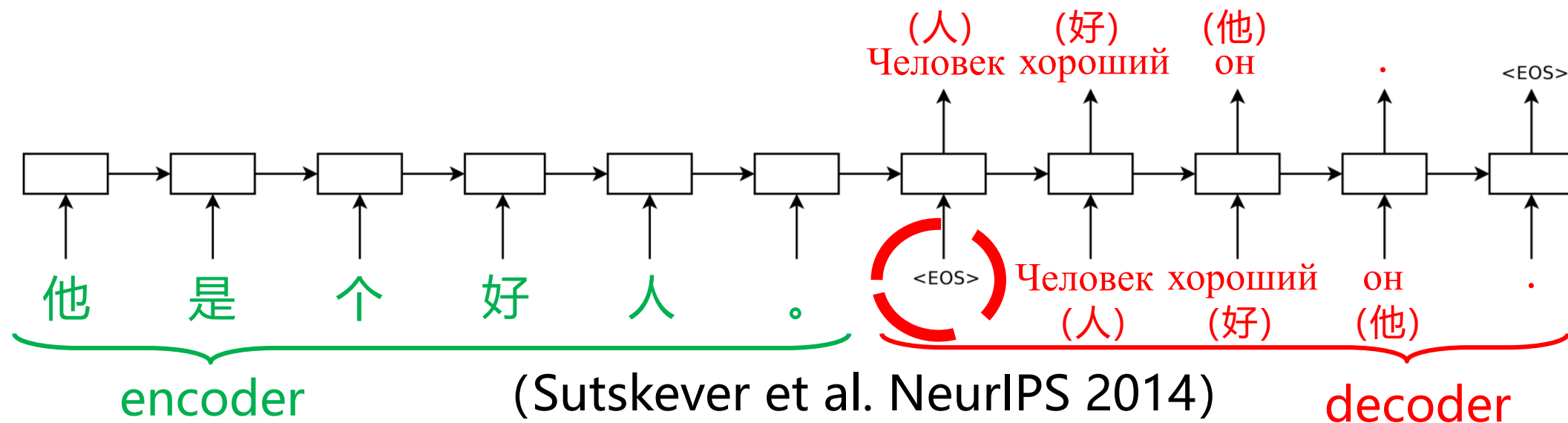
- Parallel corpus as sequence-to-sequence input.
- Rules are not necessary any more.



- ▶ **神经机器翻译**
  - ▶ 使用LSTM/GRU进行机器翻译
  - ▶ Attention机制
  - ▶ Self-attention 与Transformer模型
- ▶ **解码器 (Decoder-Only) 的大模型: GPT**
- ▶ **从GPT-3到ChatGPT: 总览**

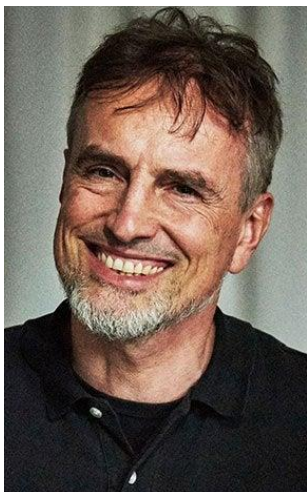
# 使用LSTM/GRU进行机器翻译

- 在神经机器翻译 (NMT) 发展的早期 (2014年), 人们直接使用LSTM来做NMT。
  - 事实上, 以机器翻译为代表的一大类任务都可以表示成这种 “先读取一段文本, 再根据这段文本生成新的文本” 的任务。这一类模型被称为seq2seq模型。
  - 通常包含一个encoder来处理源文本, 一个decoder来生成目标文本。因而也被称为encoder-decoder架构。
- 当然, 前面在序列标注中我们讲的双向模型、深层模型等变种, 都可以在这里适用。(可以怎么融合到这个场景中去?)



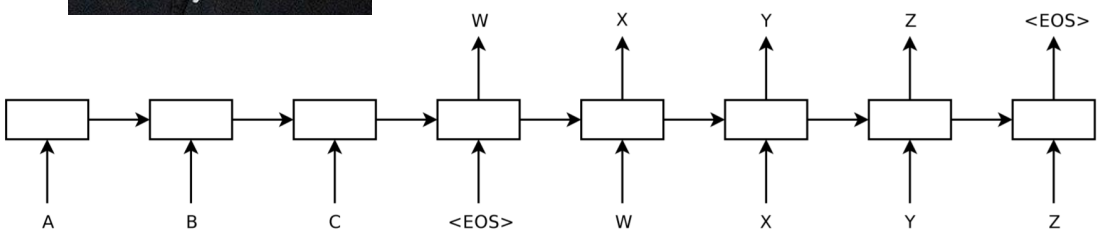
- ▶ **神经机器翻译**
  - ▶ 使用LSTM/GRU进行机器翻译
  - ▶ **Attention机制**
  - ▶ Self-attention 与Transformer模型
- ▶ **解码器 (Decoder-Only) 的大模型: GPT**
- ▶ **从GPT-3到ChatGPT: 总览**

# 2014年是一个LSTM制霸NLP的年代

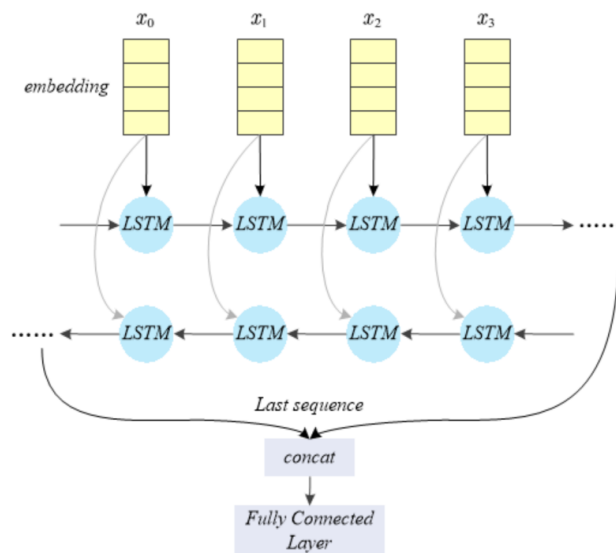


Jürgen Schmidhuber  
in 2014:

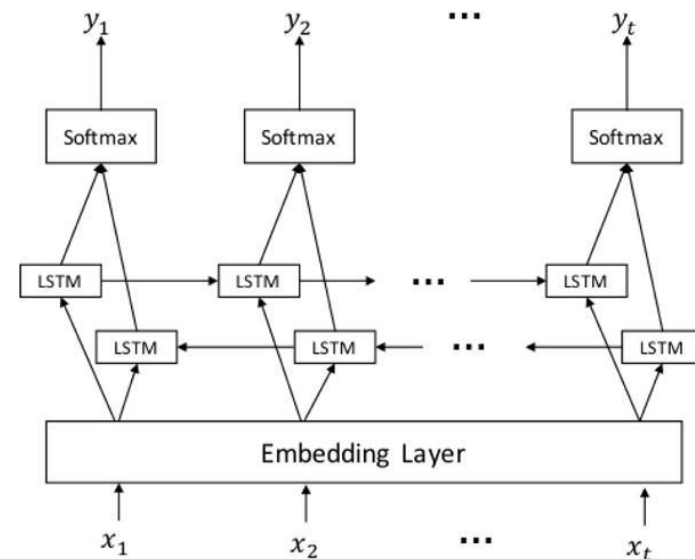
“Google itself might  
end up as one huge  
LSTM.”



机器翻译 (Sutskever et al. 2014)



文本分类/情感分析



序列标注 .....

- ▶ 看起来RNN对于序列信息处理已经很合理了，就像卷积神经网络对图片一样
- ▶ 还有什么可以进步的呢？



# LSTM/GRU等循环神经网络 (RNN) 中存在的问题

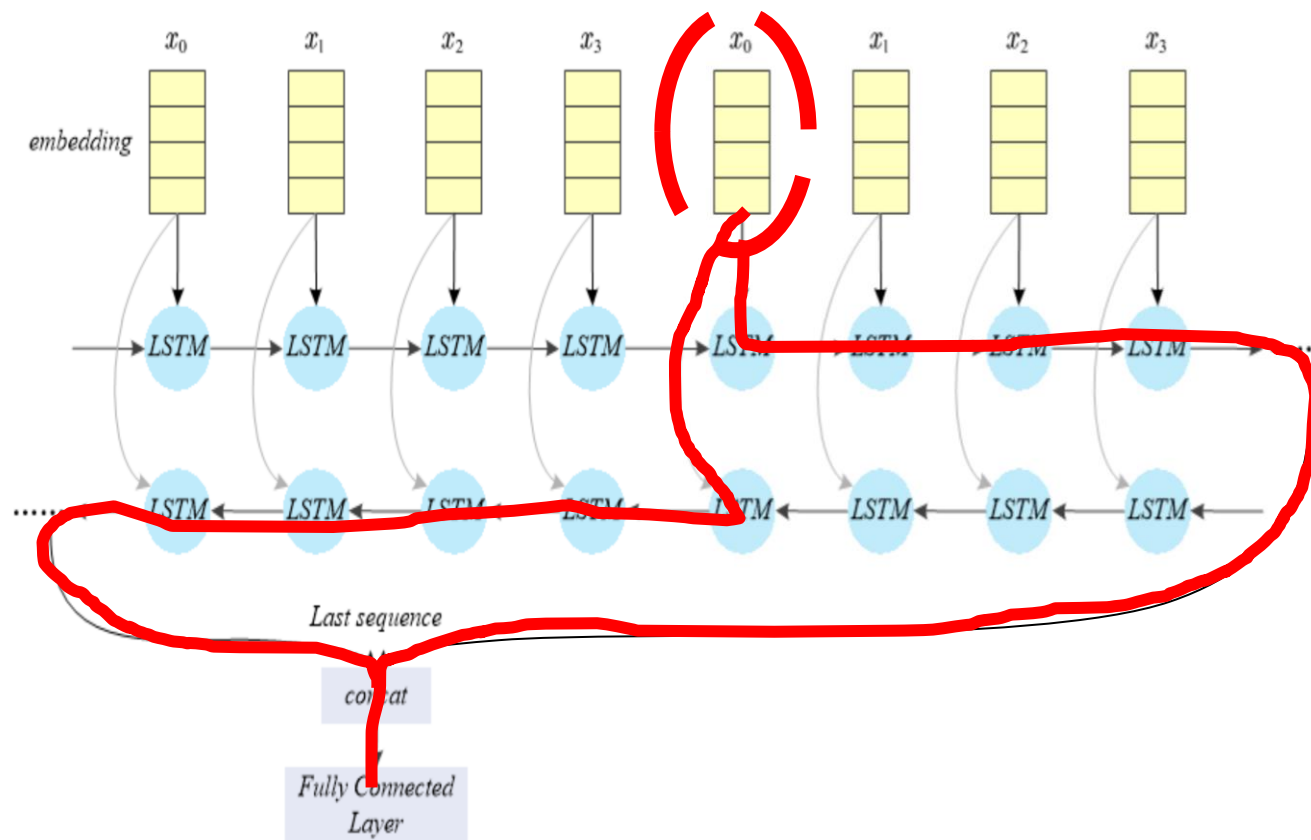
## 文本分类与情感分析:

假如序列中间的词是对分类结果产生影响的关键词

- 无论从正向还是反向, loss中的梯度传导到该词都需要漫长的路程。
- 由于RNN的梯度爆炸和梯度消失 (gradient vanishing/exploding) 问题, 这样的长程关系 (long-term dependency) 很难被有效优化。

有工作对LSTM在所有step上的h统一使用一次Max Pooling来避免该问题

- 神奇的是, 效果还真不错。。。
- 还有没有更好的解决办法?



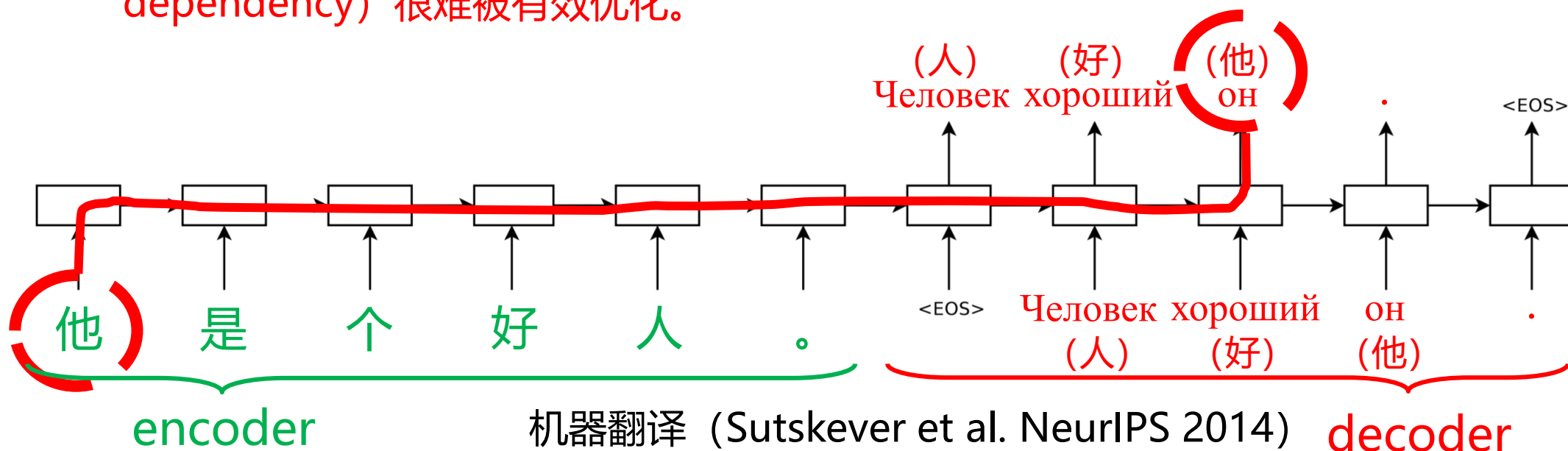
文本分类/情感分析

# LSTM/GRU等循环神经网络 (RNN) 中存在的问题

考虑机器翻译中的一个极端例子：

“他” 的语义从第一步就被存入RNN中，此后一直保存在LSTM的h和c中，直到第八步才被读取。

- 需要经过8个循环步 (recurrent step) !
- 中间不断地有新的词被存入和读取，这些新词的语义与“他”关系不大，但整个过程中“他”的语义需要被很好地保存在h和c中，占用h和c的capacity。
- 同样的，来自“он”的梯度，也需要8步才能到达“他”。由于RNN的梯度爆炸和梯度消失 (gradient vanishing/exploding) 问题，这样的长程关系 (long-term dependency) 很难被有效优化。



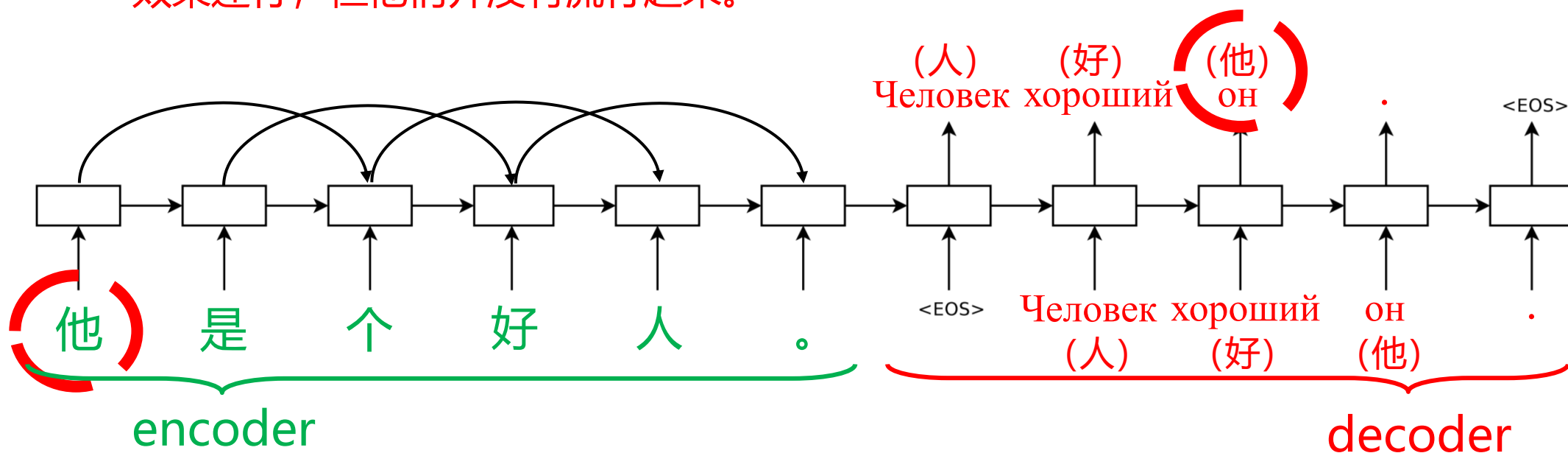
# LSTM/GRU等循环神经网络 (RNN) 中存在的问题

考虑机器翻译中的一个极端例子：

“他” 的语义从第一步就被存入RNN中，此后一直保存在LSTM的h和c中，直到第八步才被读取。

很多工作试图在RNN中添加各种各样的skip connection，来减少gradient传播所需经过的步数。

- 比如clockwork RNN (Koutník 2014), Hierarchical Multiscale RNN (Junyoung 2014)
- 效果还行，但他们并没有流行起来。



最终赢得这方面尝试的成功方案，是注意力机制 ( Attention Mechanism )



事实上，人在翻译长句子的某一个词的时候，更关心这个词对应的原文中相关的几个词，而并不关心别的上下文。

也就是说，只要能正确找到当前词对应的原文 (learning to align) ，翻译就成功了一大半。

体会一下：

The common belief of some linguists that each language is a perfect vehicle for the thoughts of the nation speaking it is in some ways the exact counterpart of the conviction of the Manchester school of economics that supply and demand will regulate everything for the best.

怎么在神经网络中体现？

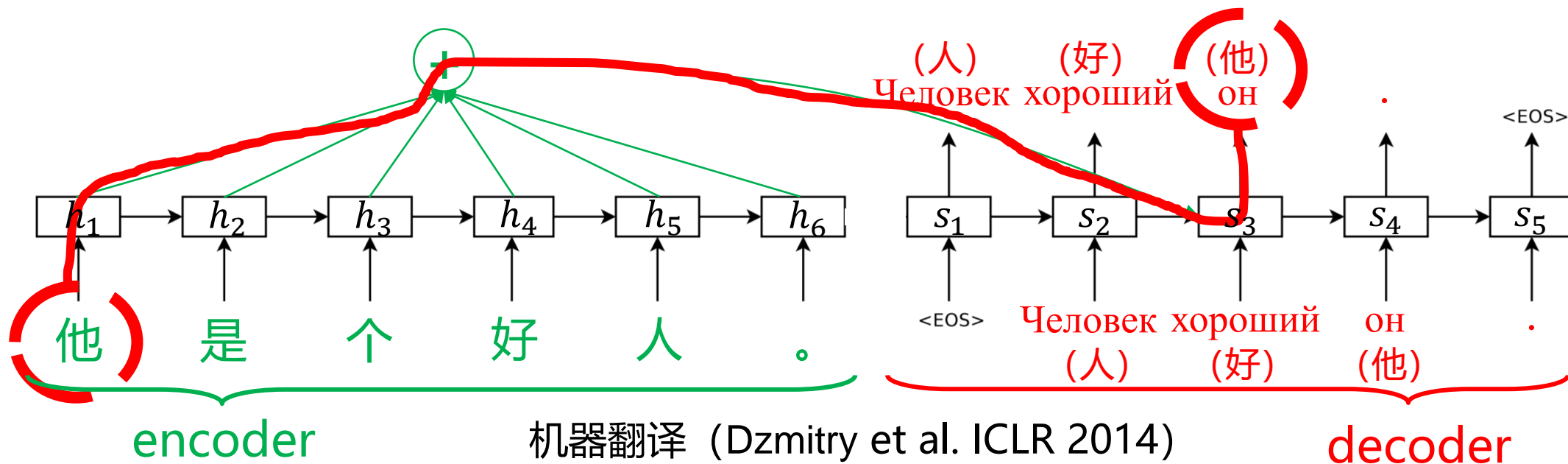
# 注意力机制 ( Attention Mechanism )

考虑机器翻译中的一个极端例子：

~~“他” 的语义从第一步就被存入RNN中，此后一直保存在LSTM的h和c中，直到第八步才被读取。~~

Decoder在预测“он”时，不光使用前一步的RNN state ( $s_{i-1}$ )，而且使用encoder的所有RNN state (即 $h_1 \sim h_N$ ) 的线性组合。

每个 $h_i$ 处所分到的线性组合权重由 $s_{i-1}$  与 $h_i$ 共同决定。



# 注意力机制 ( Attention Mechanism )

## 普通RNN

$$s_i = f(s_{i-1}, y_{i-1})$$

$f$ 是任意非线性函数，  
代表RNN中一步的运算。

## 带有attention的RNN

$$s_i = f(s_{i-1}, y_{i-1}, c_i)$$

$$c_i = \sum_{j=1}^N \alpha_{ij} h_j$$

$$\alpha_{ij} = \text{softmax} \frac{\exp(e_{ij})}{\sum_{k=1}^N \exp(e_{ik})}$$

$$e_{ij} = g(s_{i-1}, h_j)$$

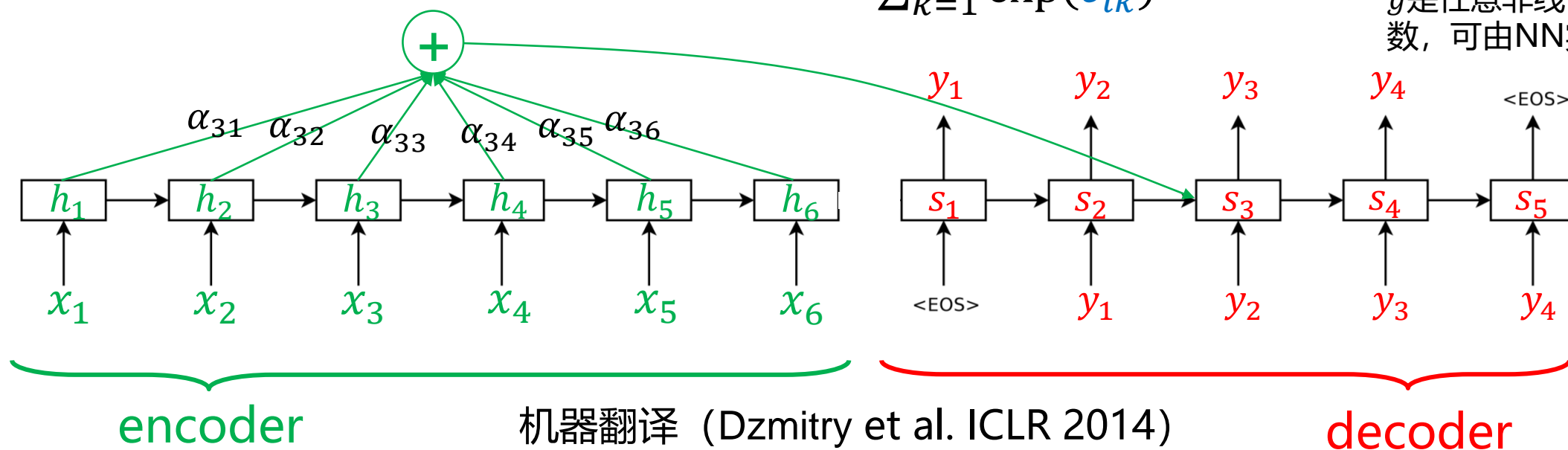
$g$ 是任意非线性函数，  
可由NN实现。



Dzmitry Bahdanau



Yoshua Bengio



# 注意力机制 ( Attention Mechanism )

更多细节:

- Dzmitry的原文中encoder使用的是BiGRU。
- $c_i$ 在decoder的GRU内部, 像 $s_{i-1}$ 一样, 乘上一个矩阵后与其他两个元素相加。
- G的实现是一个两层的全连接NN, tanh作为隐层激活函数。 (还可以用什么?)

带有attention的RNN

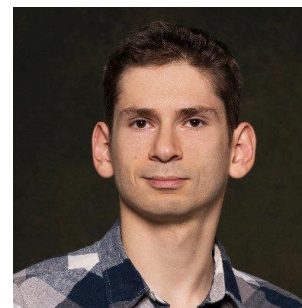
$$s_i = f(s_{i-1}, y_{i-1}, c_i)$$

$$c_i = \sum_{j=1}^N \alpha_{ij} h_j$$

$$\alpha_{ij} = \text{softmax} \frac{\exp(e_{ij})}{\sum_{k=1}^N \exp(e_{ik})}$$

$$e_{ij} = g(s_{i-1}, h_j)$$

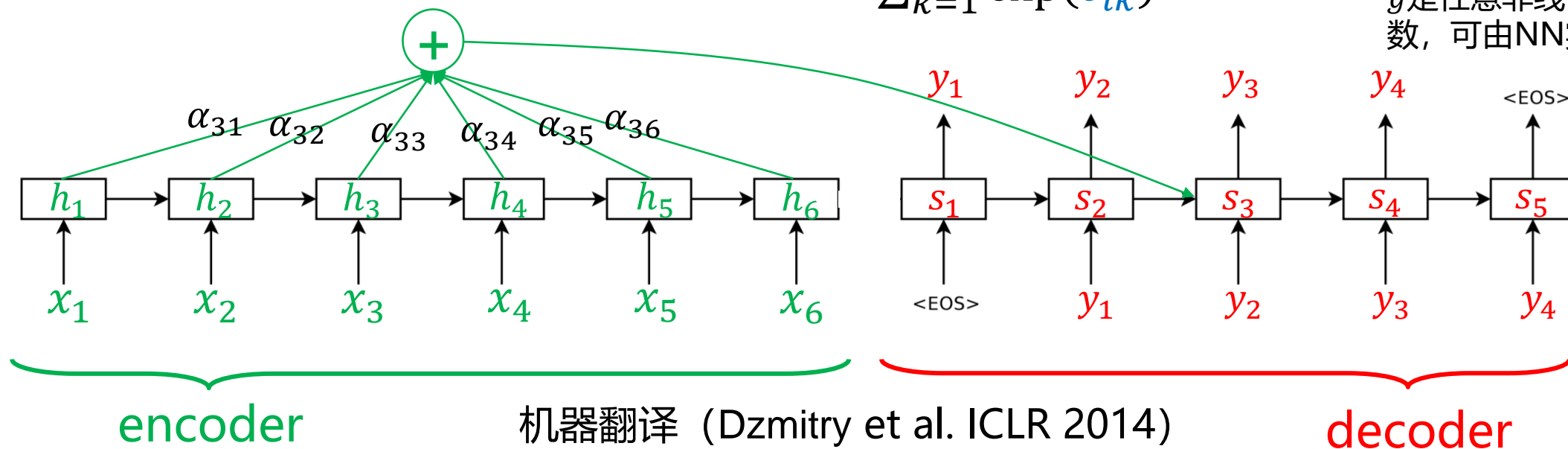
$g$ 是任意非线性函数, 可由NN实现。



Dzmitry Bahdanau



Yoshua Bengio



- ▶ **神经机器翻译**
  - ▶ 使用LSTM/GRU进行机器翻译
  - ▶ Attention机制
  - ▶ **Self-attention** 与Transformer模型
- ▶ **解码器 (Decoder-Only) 的大模型: GPT**
- ▶ **从GPT-3到ChatGPT: 总览**



- ▶ **Attention用于机器翻译取得了巨大的成功。**
- ▶ **在没有decoder的时候，比如做文本分类、情感分析的时候，如何使用attention？**

# 自注意力机制 (Self-attention)

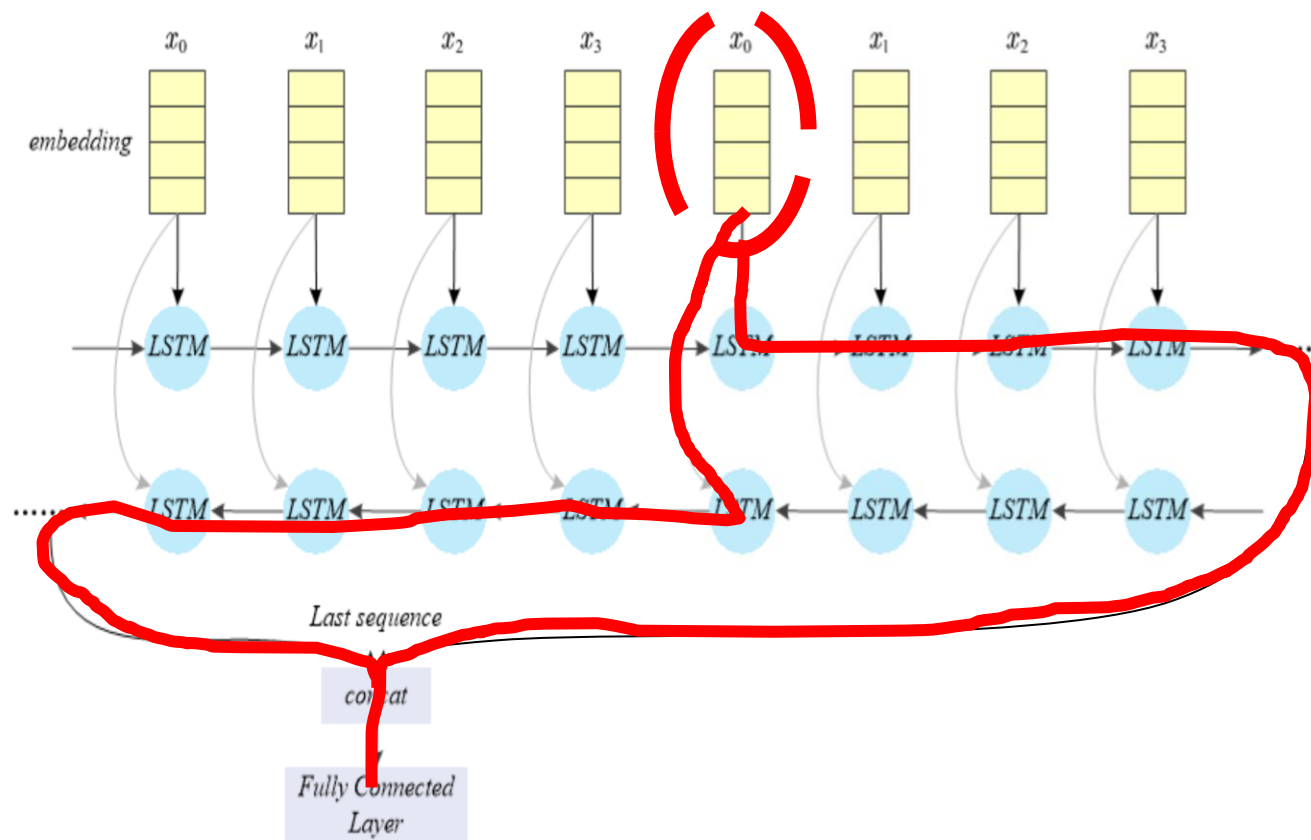
## 文本分类与情感分析:

假如序列中间的词是对分类结果产生影响的关键词

- 无论从正向还是反向, loss中的梯度传导到该词都需要漫长的路程。
- 由于RNN的梯度爆炸和梯度消失 (gradient vanishing/exploding) 问题, 这样的长程关系 (long-term dependency) 很难被有效优化。

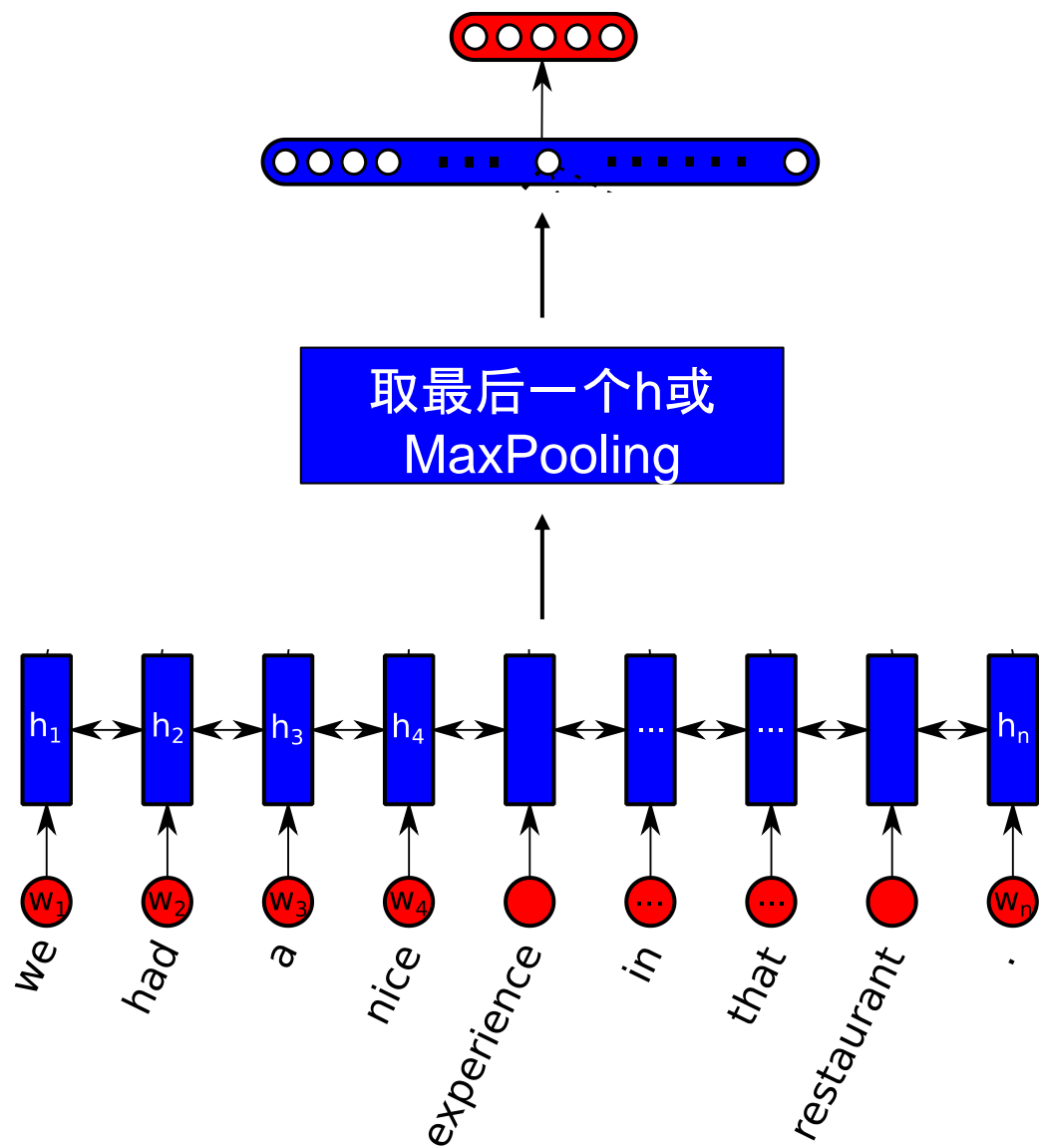
有工作对LSTM在所有step上的h统一使用一次Max Pooling来避免该问题

- 神奇的是, 效果还真不错。。。
- 还有没有更好的解决办法?



文本分类/情感分析

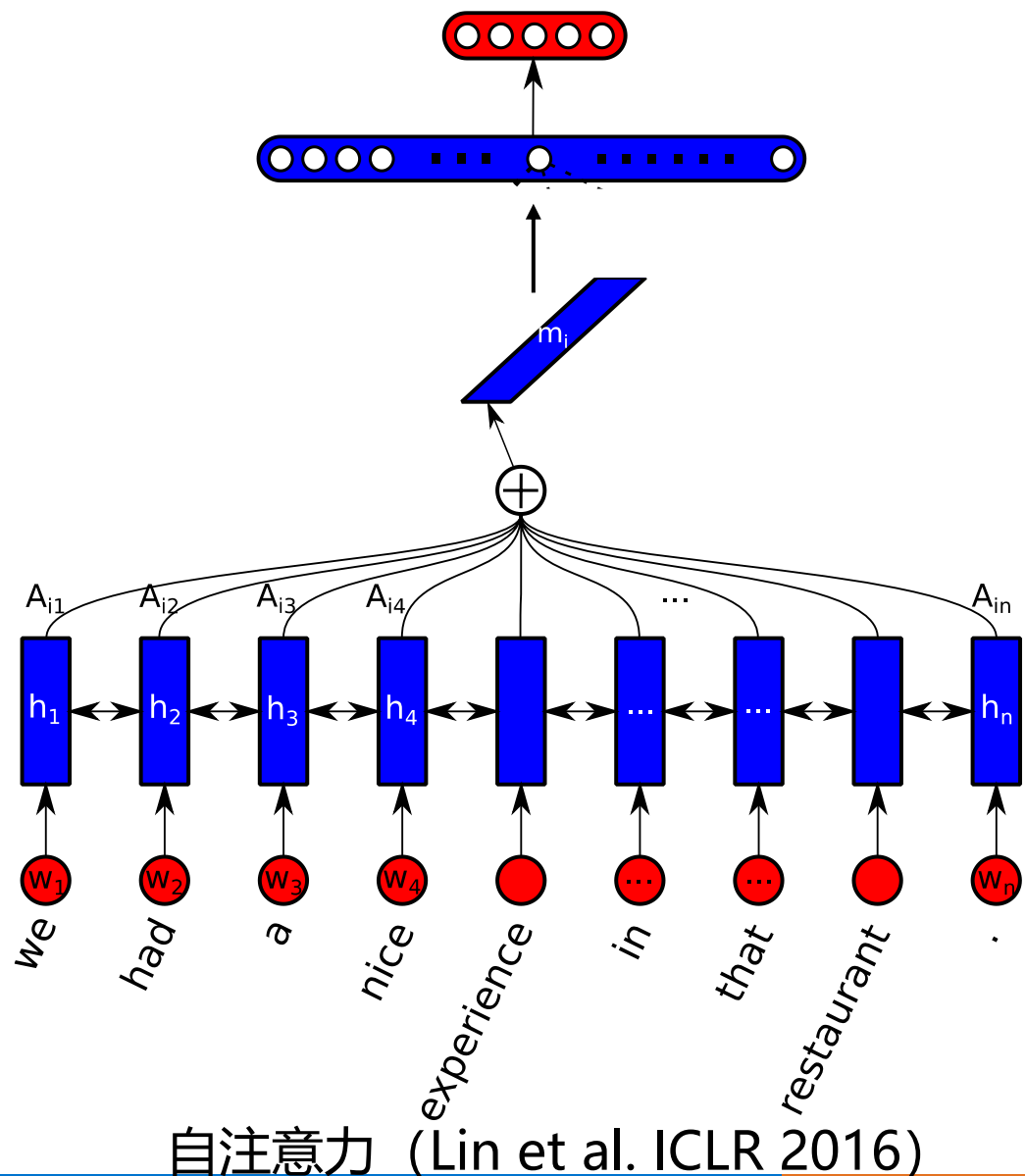
# 自注意力机制 (Self-attention)



以情感分析为例，将一个句子分为5个不同的类别。

左边是attention出现之前常用的情感分析模型。

# 自注意力机制 (Self-attention)



“取最后一个h或MaxPooling” → “每个 $h_i$ 的加权求和”

加权的权值 $A_j$ 由一个两层的全连接NN输出，再通过softmax得到。

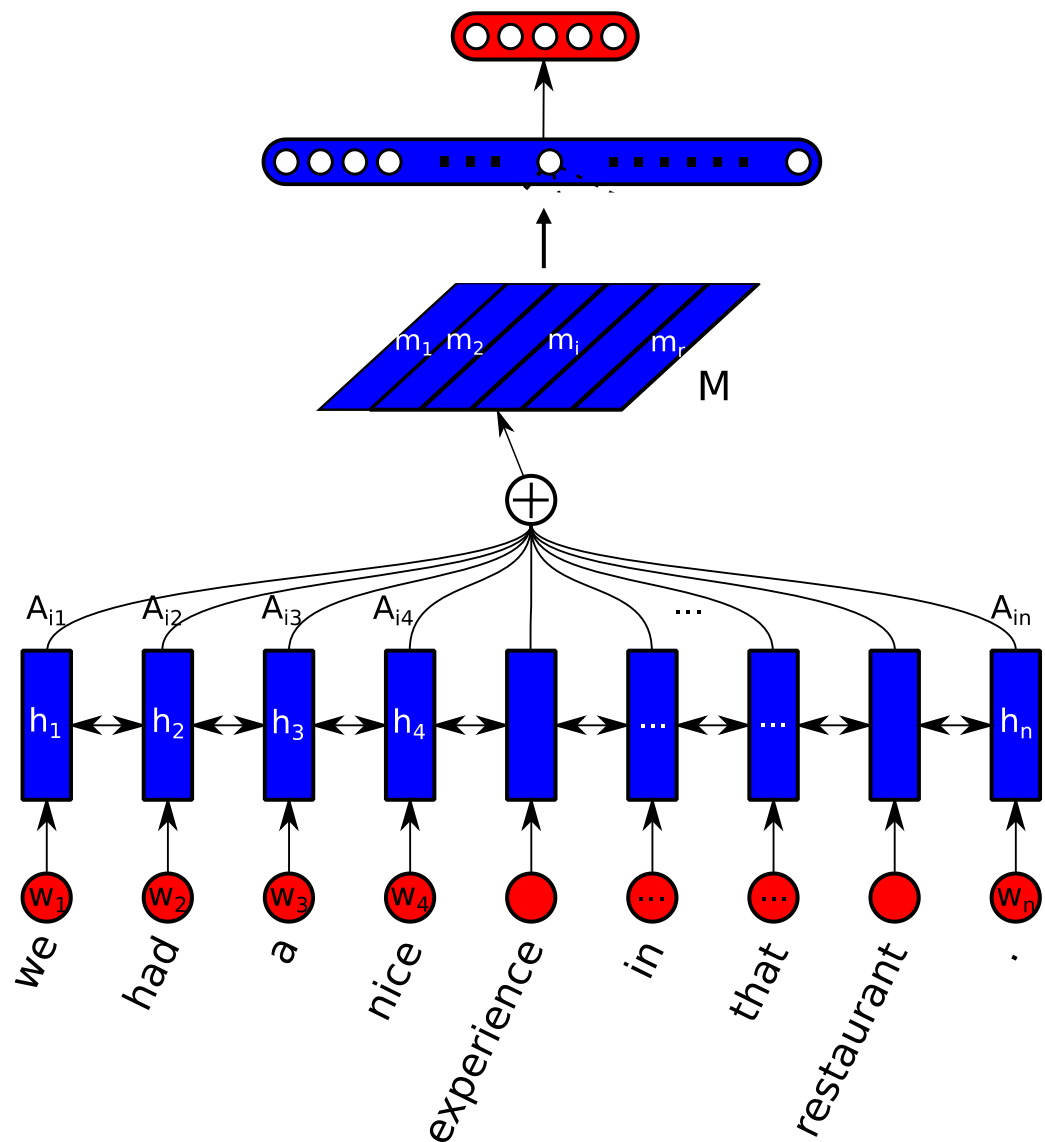
$$A_j = \frac{\exp(e_j)}{\sum_{k=1}^N \exp(e_k)}$$

$$e_j = g(h_1, h_2, \dots, h_N)$$

$$= W_2 \tanh(W_1 [h_1, h_2, \dots, h_N])$$

自注意力 (Lin et al. ICLR 2016)

# 自注意力机制 (Self-attention)



自注意力 (Lin et al. ICLR 2016)

“取最后一个h或MaxPooling” → “每个 $h_i$ 的加权求和”

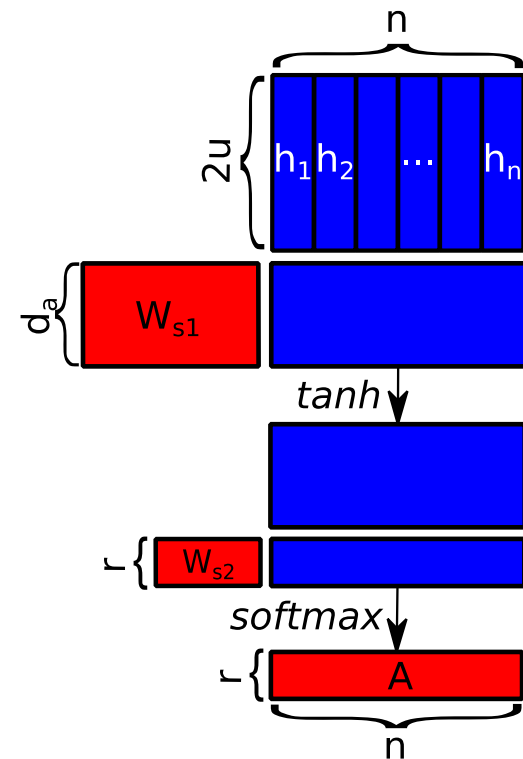
加权的权值 $A_j$ 由一个两层的全连接NN输出, 再通过softmax得到。

多头自注意力: 不止计算一个加权平均, 而是同时计算多个。

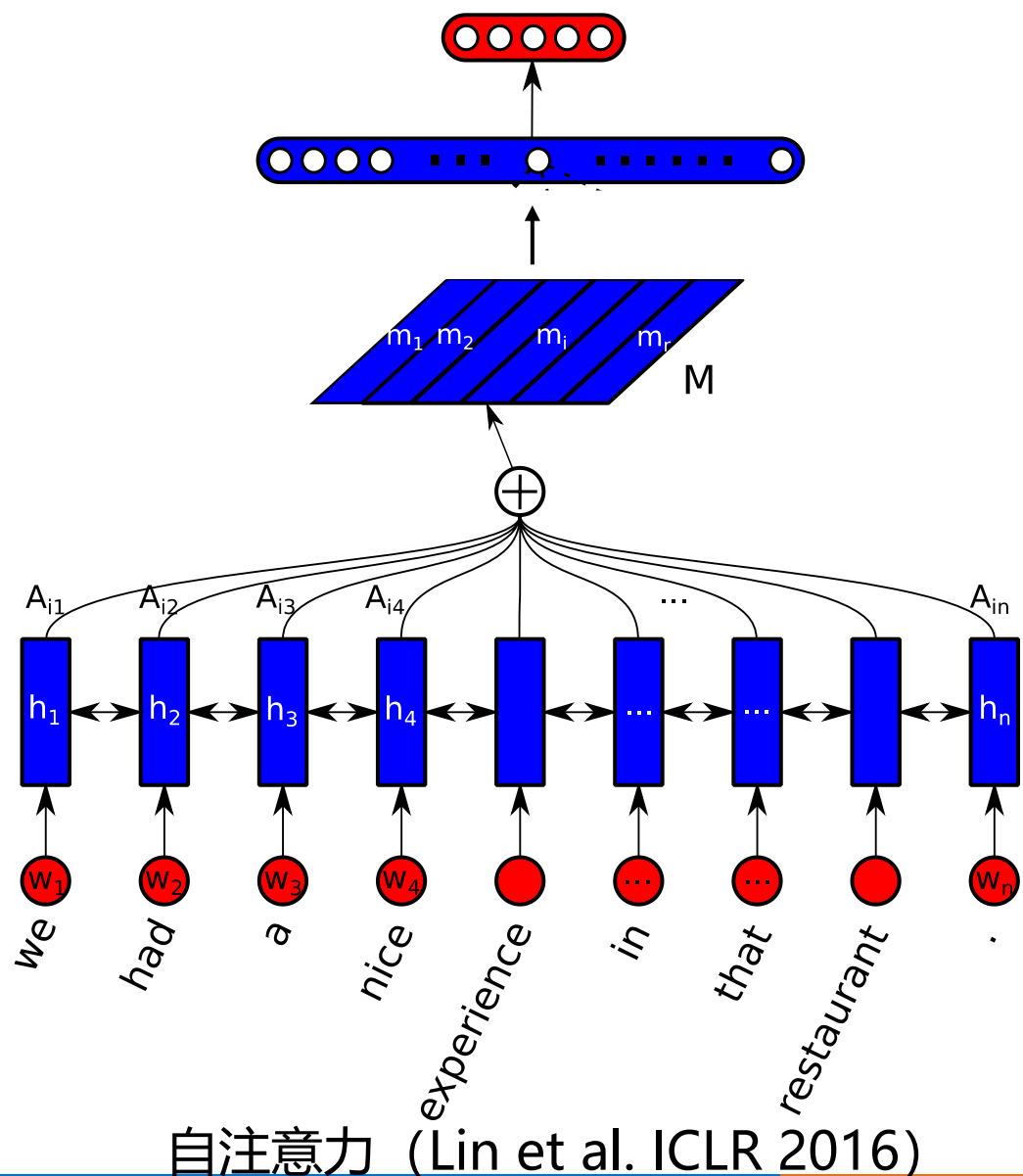
$$A_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^N \exp(e_{ik})}$$

$$e_{ij} = g(h_1, h_2, \dots, h_N)$$

$g$ 的具体形式 →



# 自注意力机制 (Self-attention)



“取最后一个h或MaxPooling” → “每个 $h_i$ 的加权求和”

加权的权值 $A_j$ 由一个两层的全连接NN输出，再通过softmax得到。

**多头自注意力**：不止计算一个加权平均，而是同时计算多个。

更进一步的，如果我们作以下三方面改变：

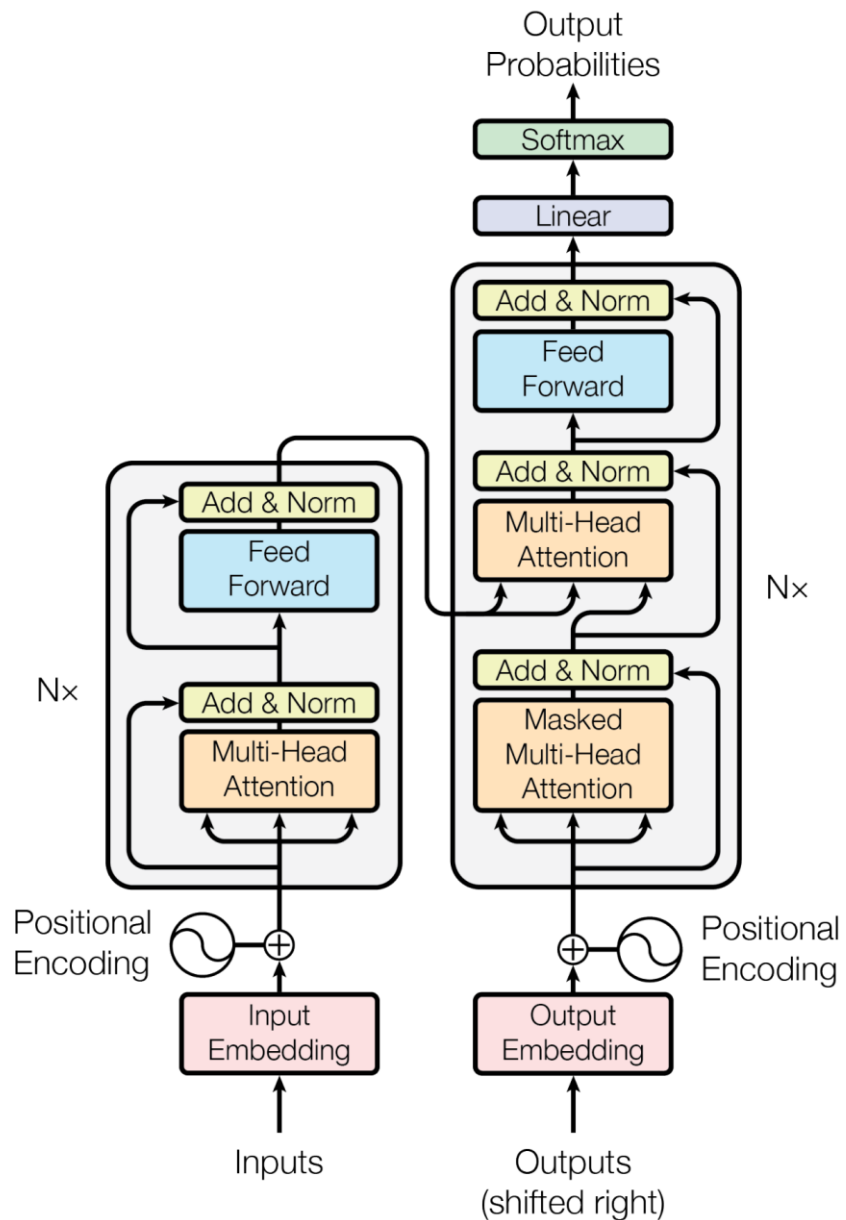
1. 干脆撤掉底下的RNN，
2. 对每一个 $h_i$ 计算一组多头自注意力机制所得到的向量集，
3. 利用额外的positional embedding弥补撤掉RNN所引起的位置信息缺失

那么连RNN都可以不需要了，整个模型只使用了attention。我们就得到了Transformer。

自注意力 (Lin et al. ICLR 2016)

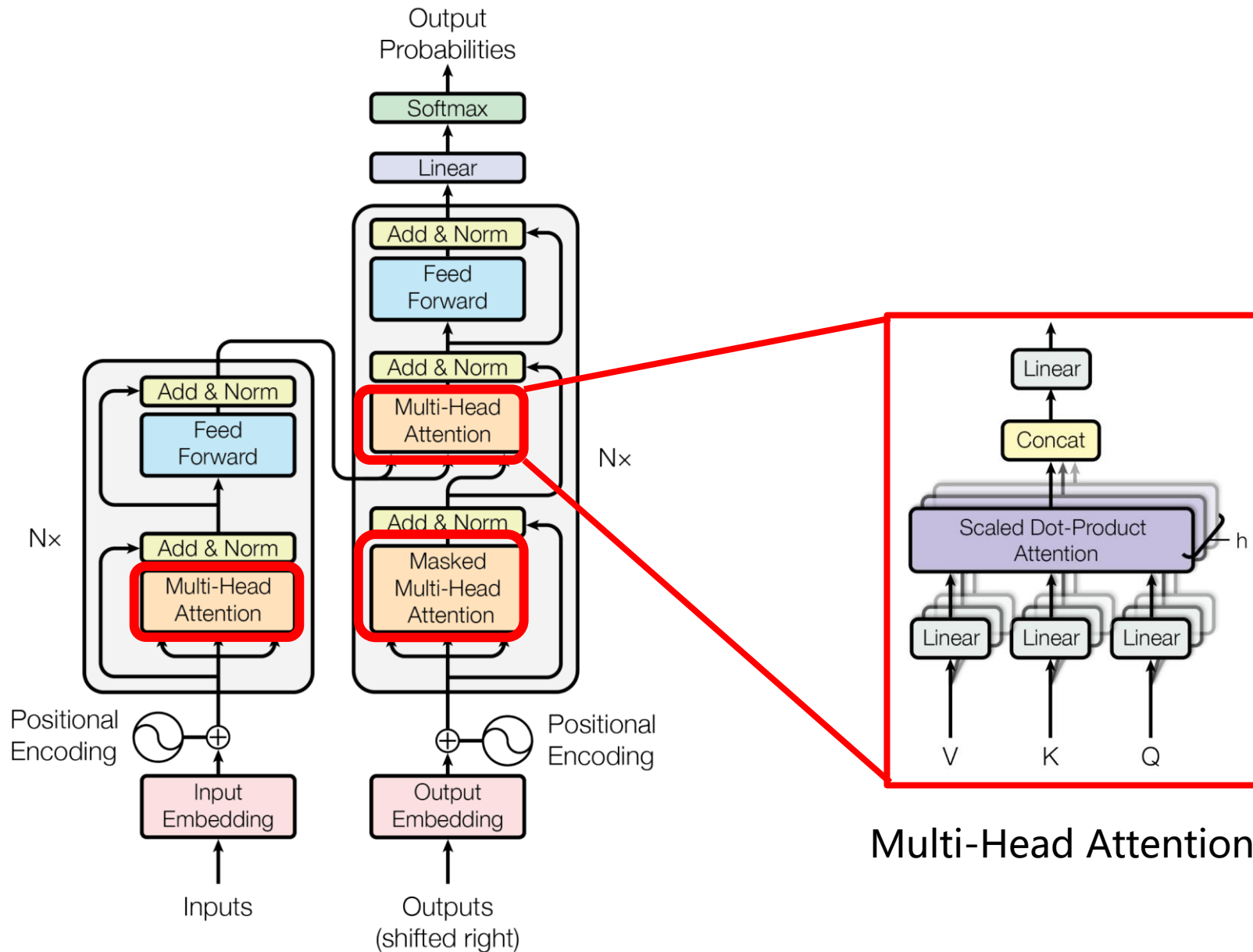
- ▶ **神经机器翻译**
  - ▶ 使用LSTM/GRU进行机器翻译
  - ▶ Attention机制
  - ▶ Self-attention 与Transformer模型
- ▶ **解码器 (Decoder-Only) 的大模型: GPT**
- ▶ **从GPT-3到ChatGPT: 总览**

# 整体框架

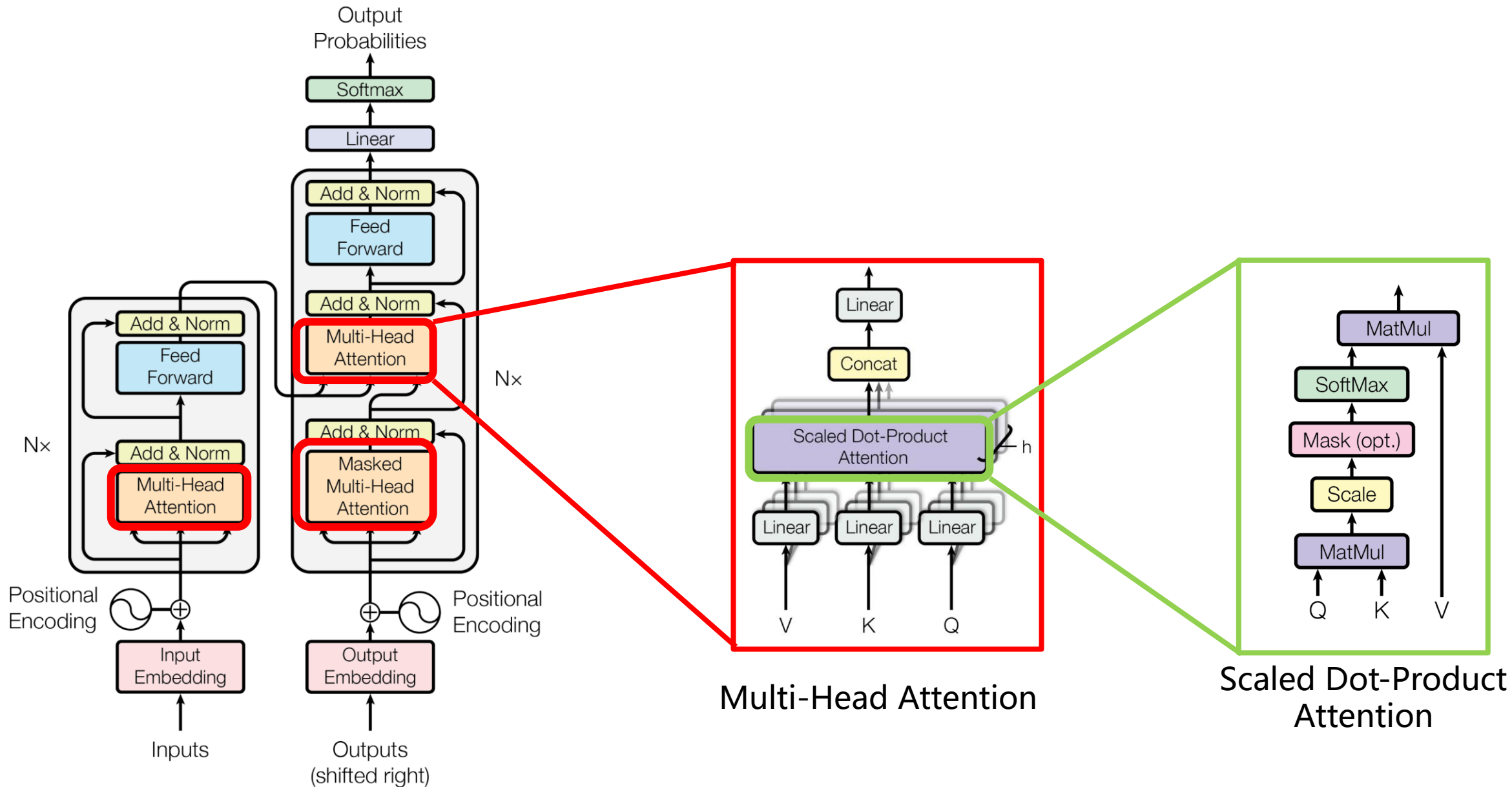




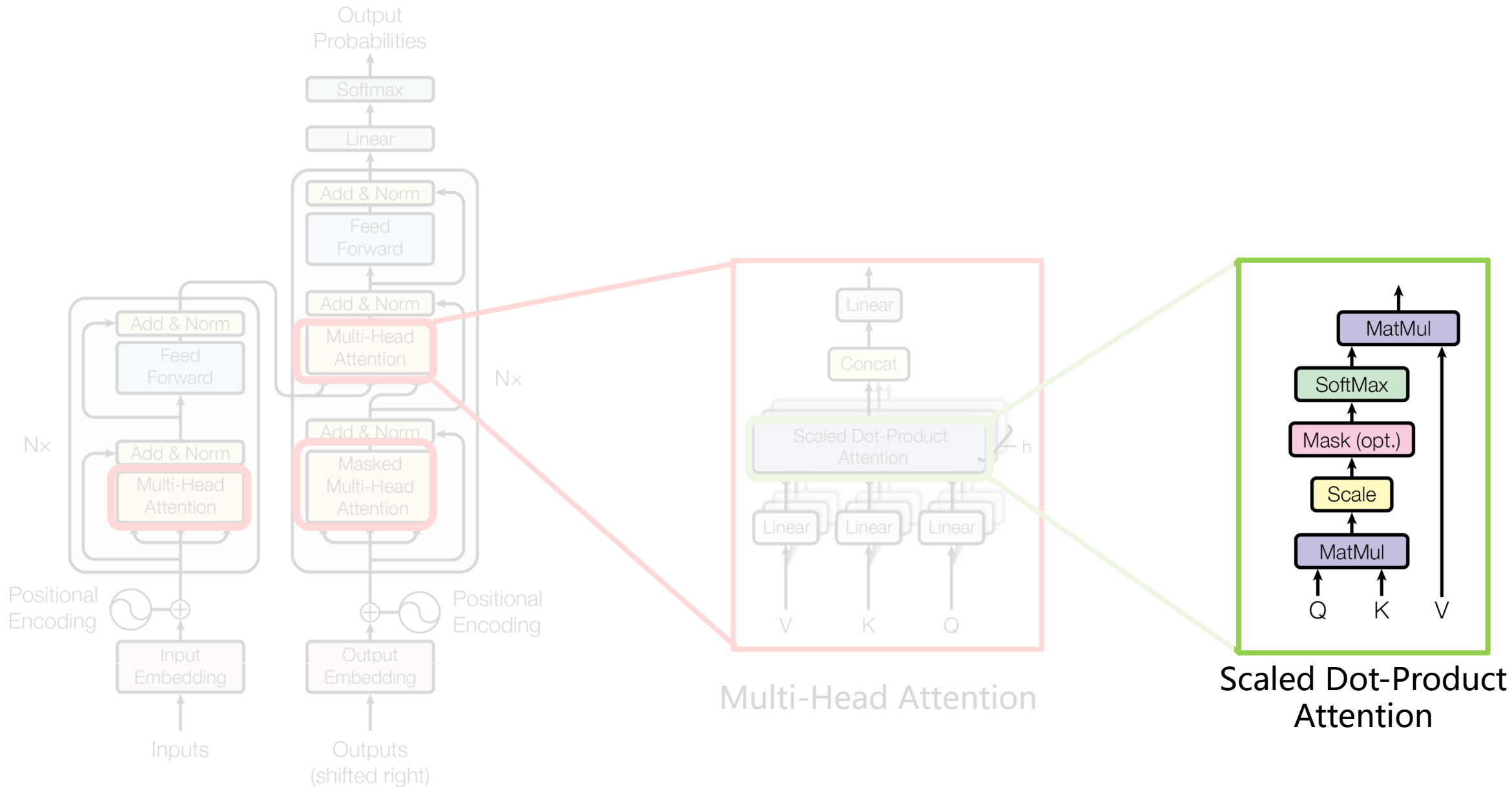
# 整体框架



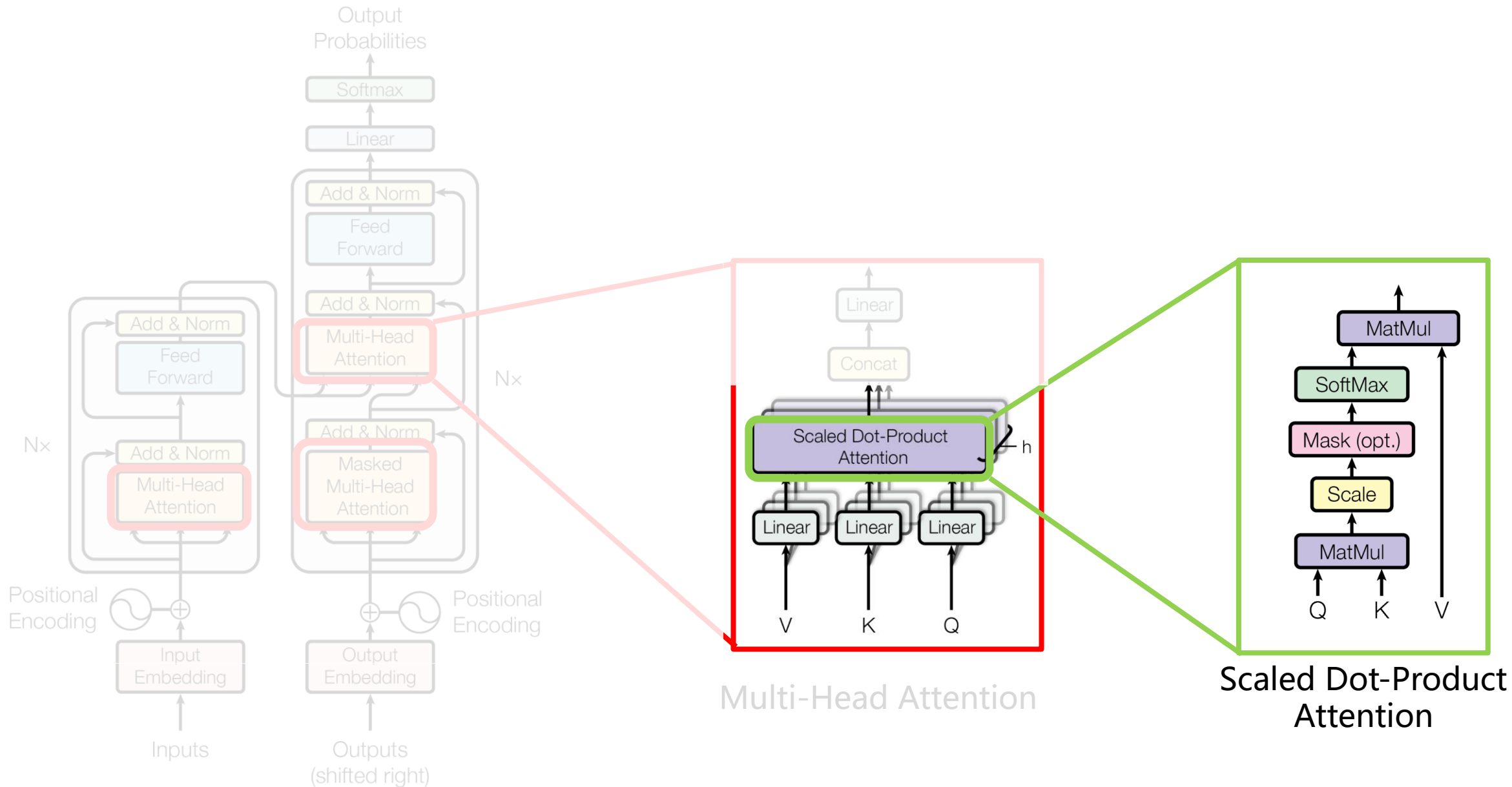
# 整体框架



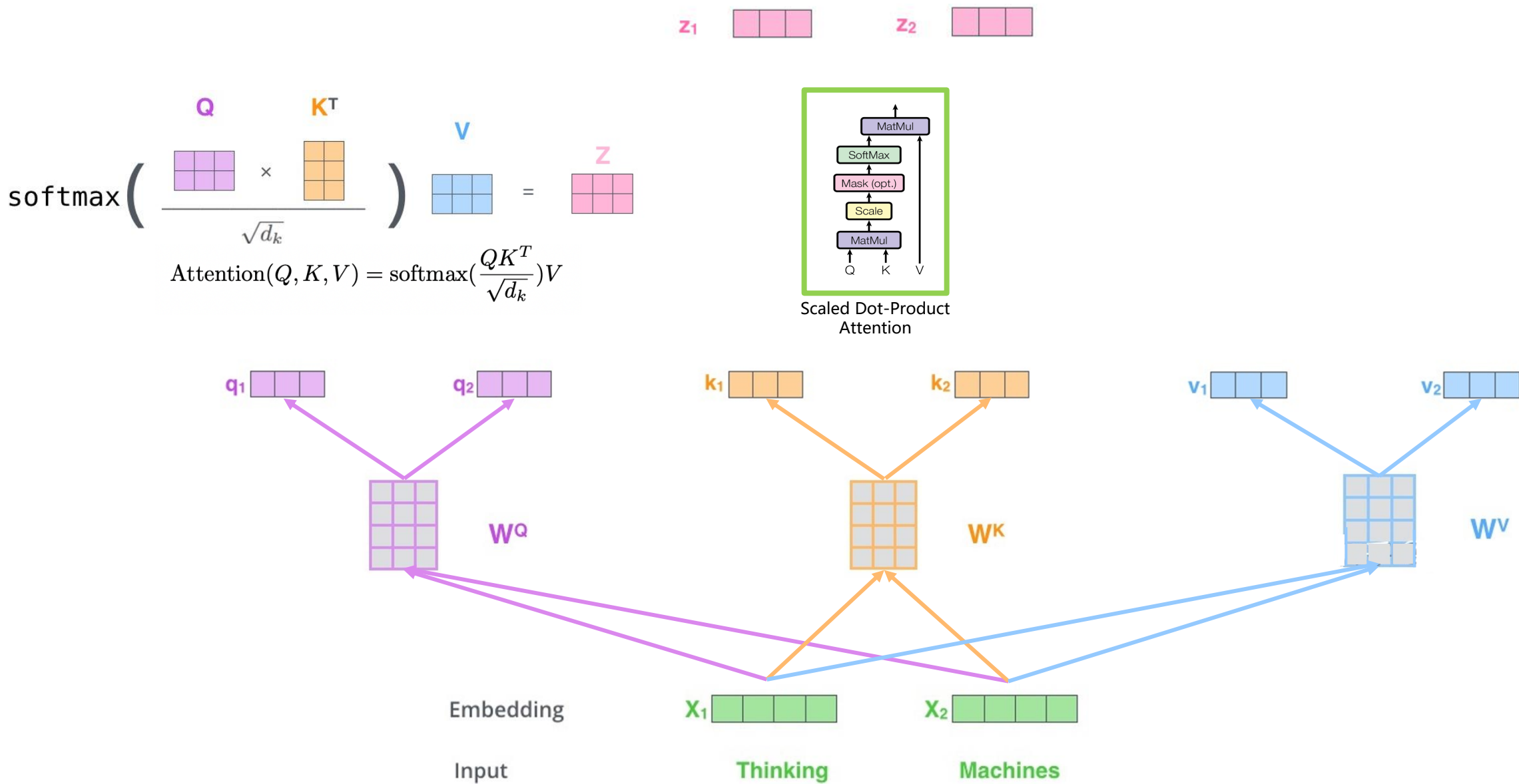
# 整体框架



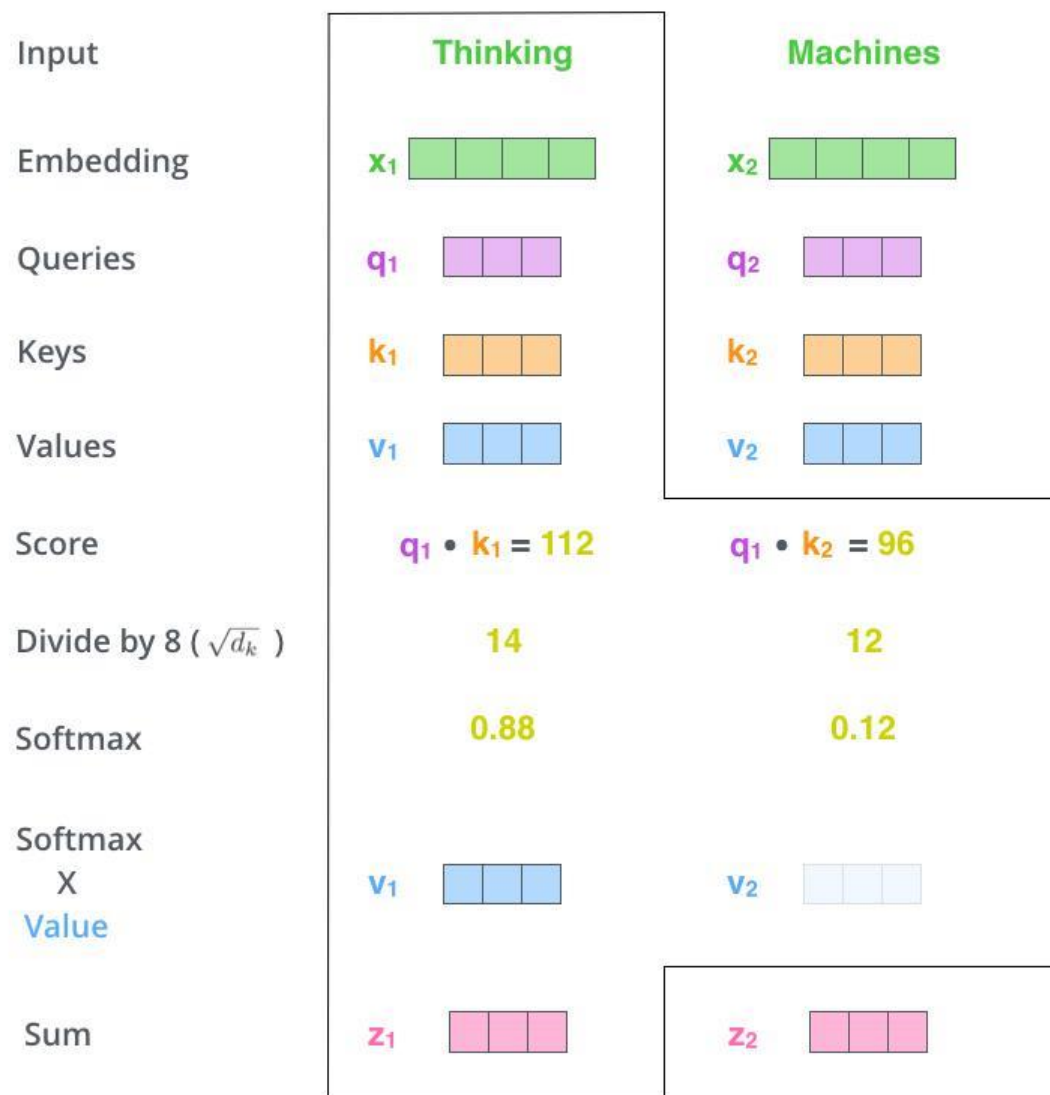
# 整体框架



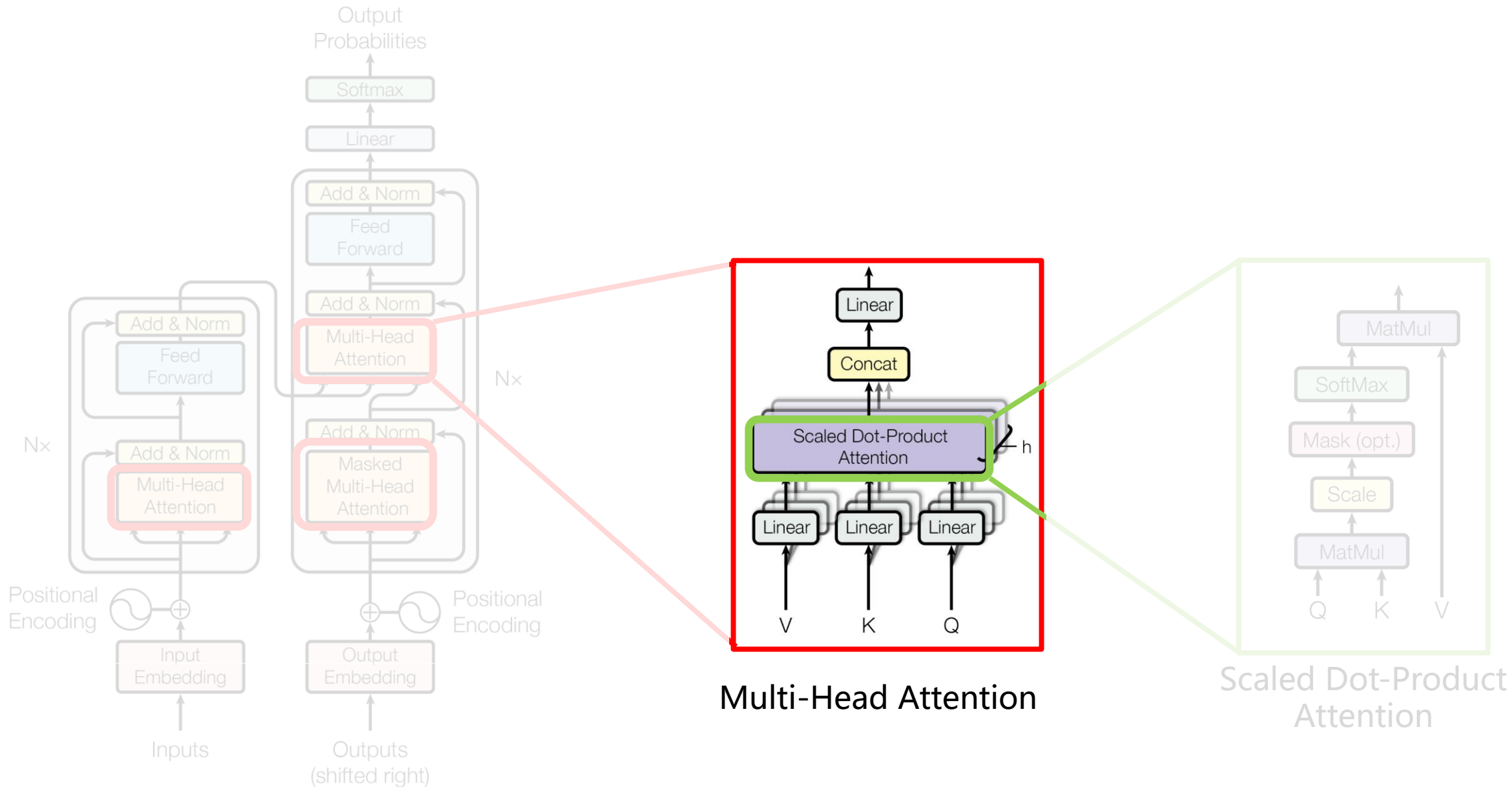
# Q, K, V and Scaled Dot-Product Attention



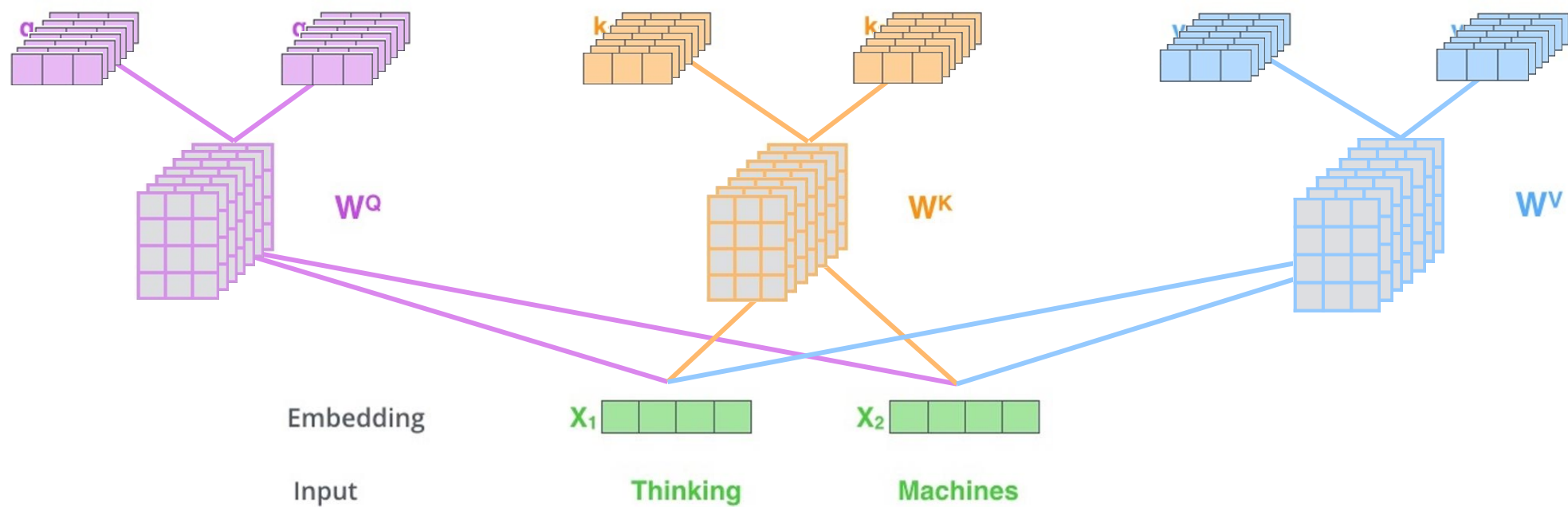
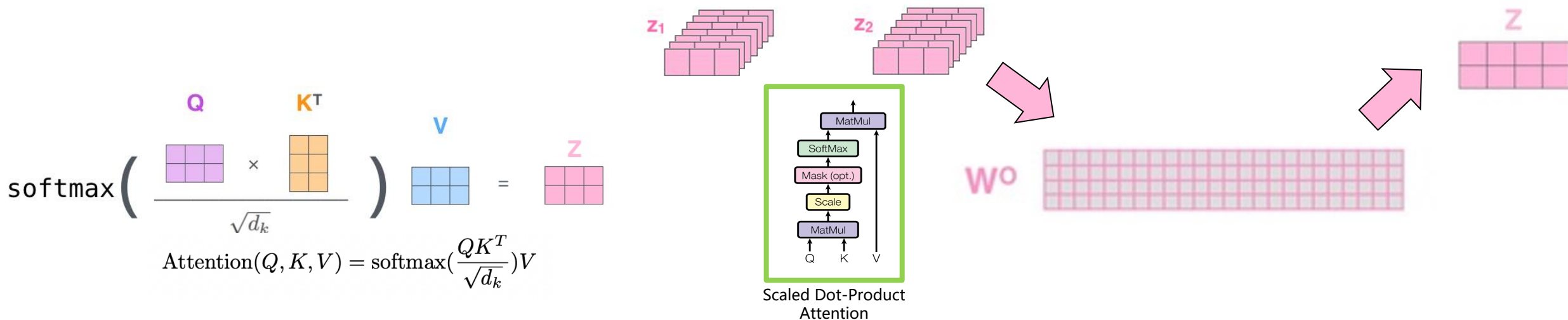
# Q, K, V and Scaled Dot-Product Attention



# 整体框架

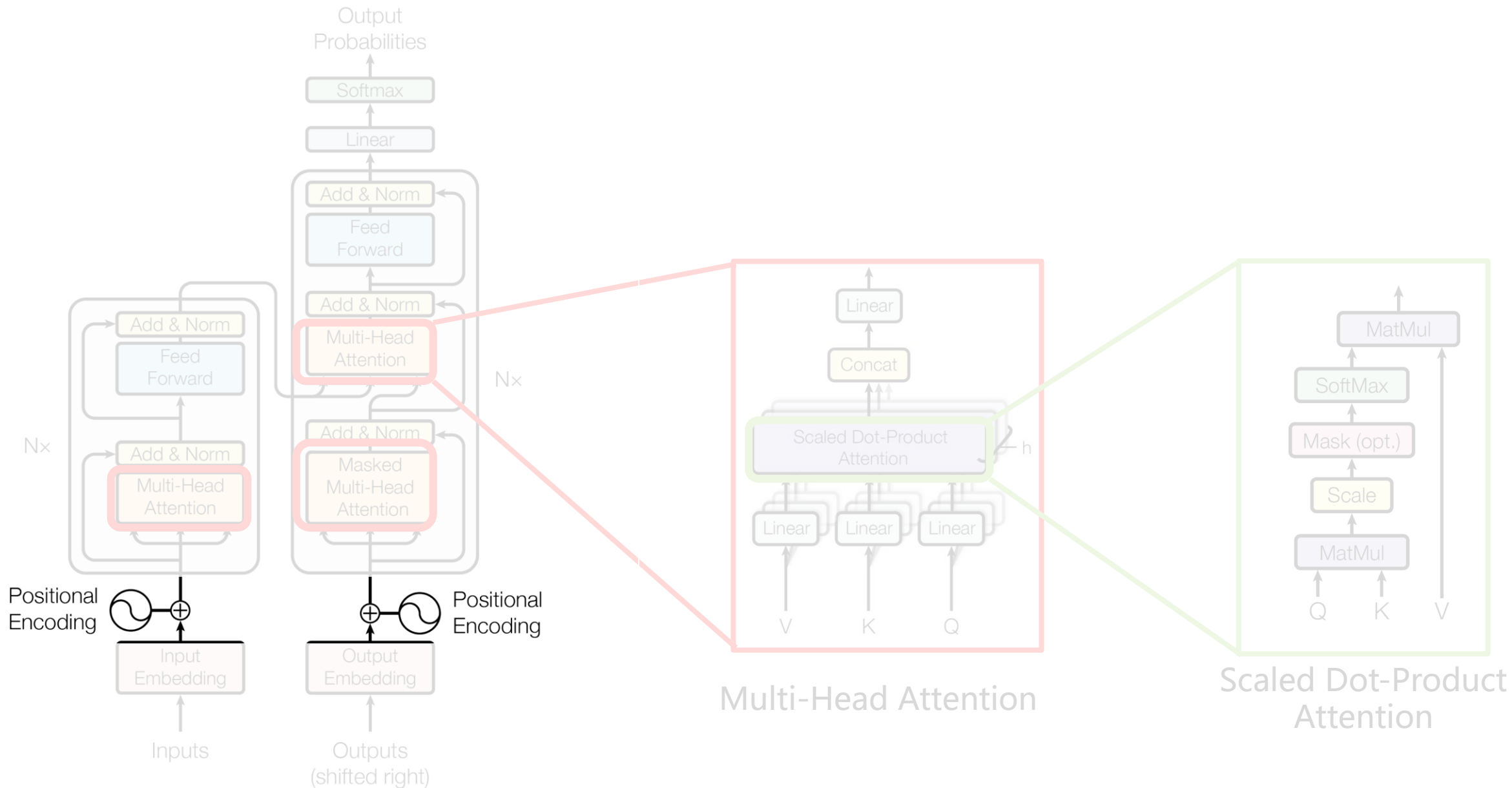


# Q, K, V and Scaled Dot-Product Attention with Multiple Heads

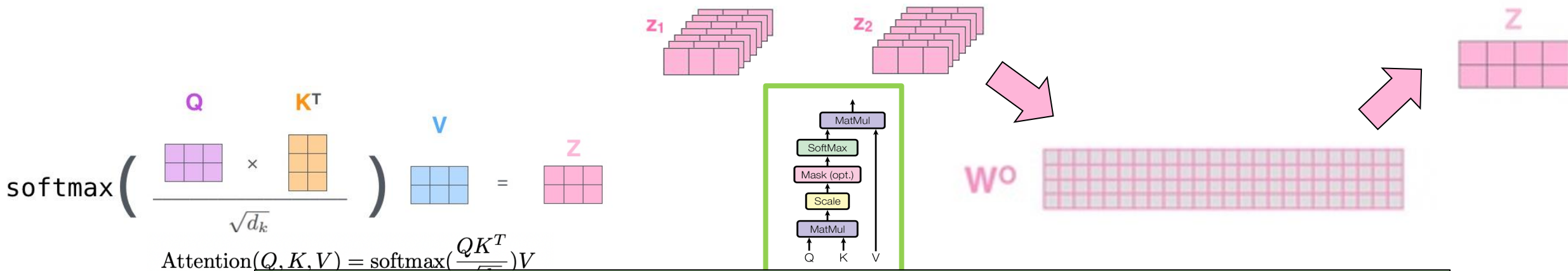




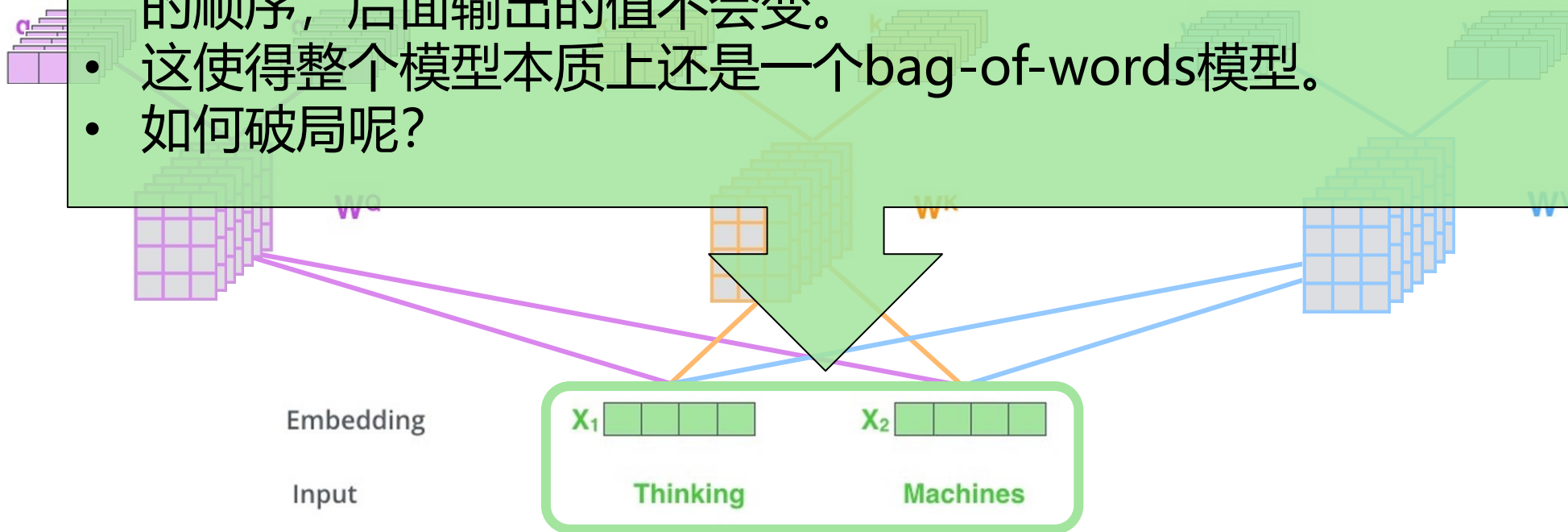
# 整体框架



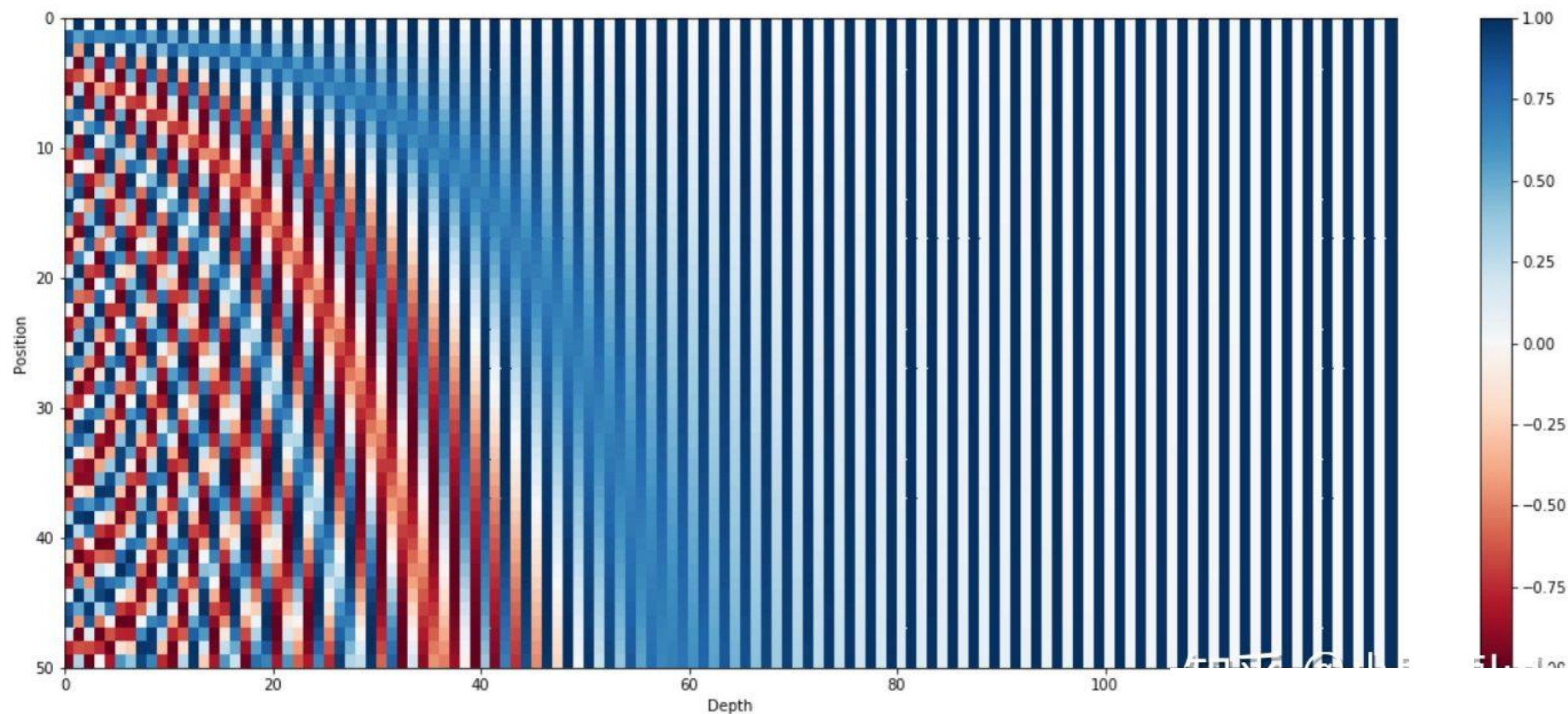
# Q, K, V and Scaled Dot-Product Attention with Multiple Heads



- 词序没有被考虑进来。也就是说，我们可以任意调换输入单词的顺序，后面输出的值不会变。
- 这使得整个模型本质上还是一个bag-of-words模型。
- 如何破局呢？



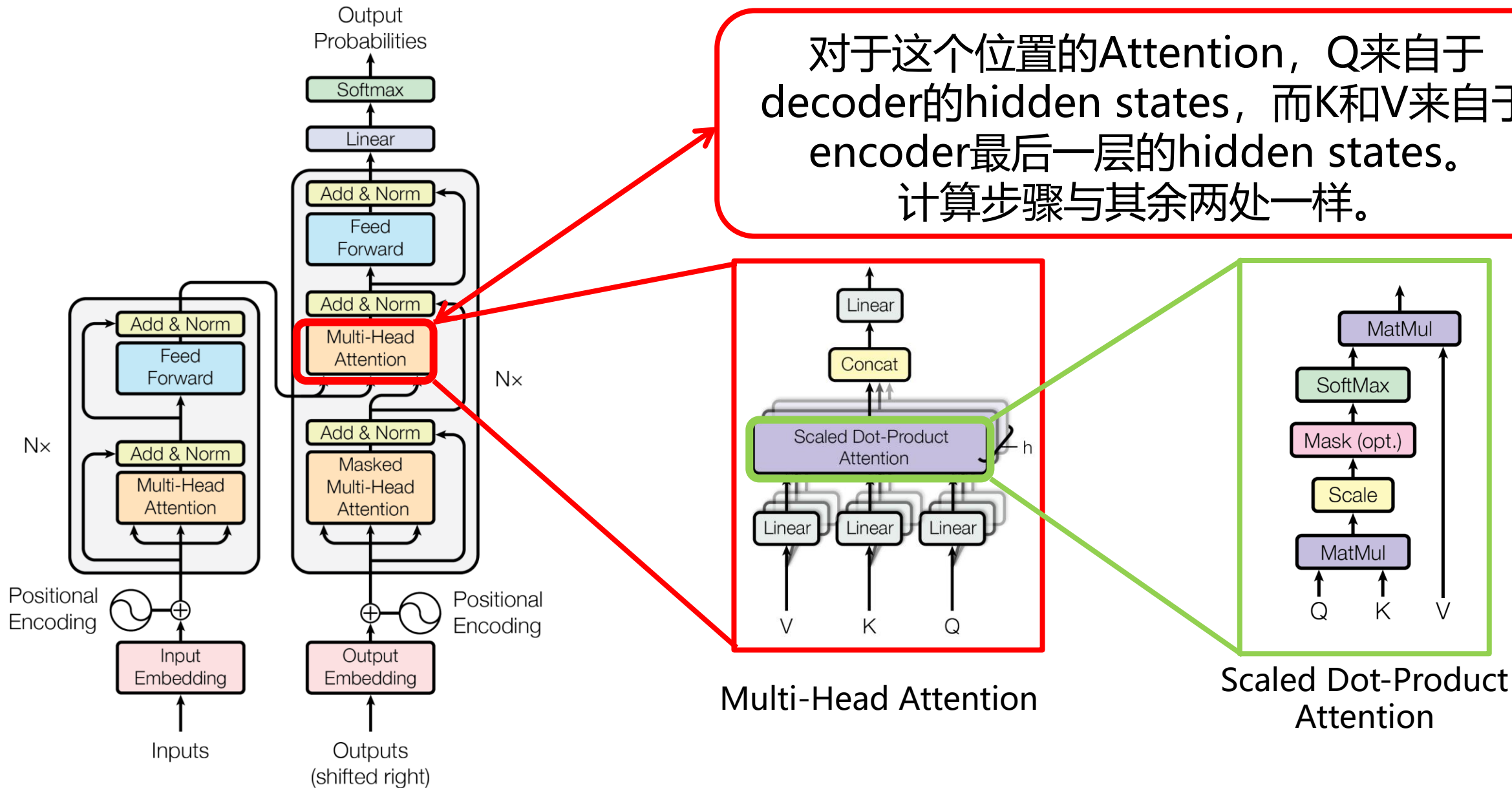
# Position Encoding (PE)



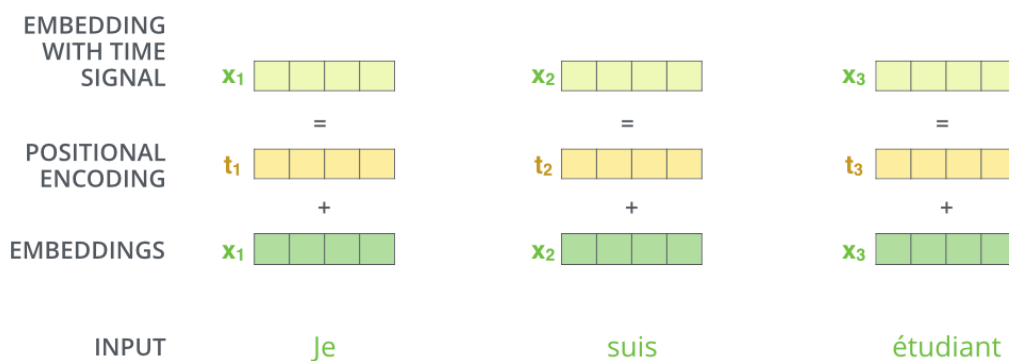
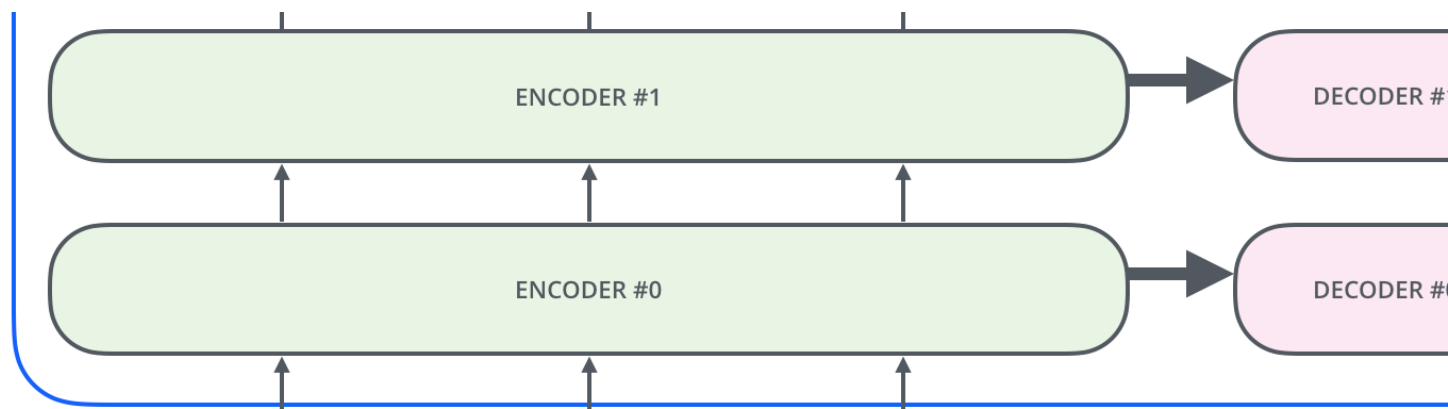
$$\mathbf{pe}_{(j,2i)} = \sin(j/10000^{2i/d_{model}}),$$

$$\mathbf{pe}_{(j,2i+1)} = \cos(j/10000^{2i/d_{model}}),$$

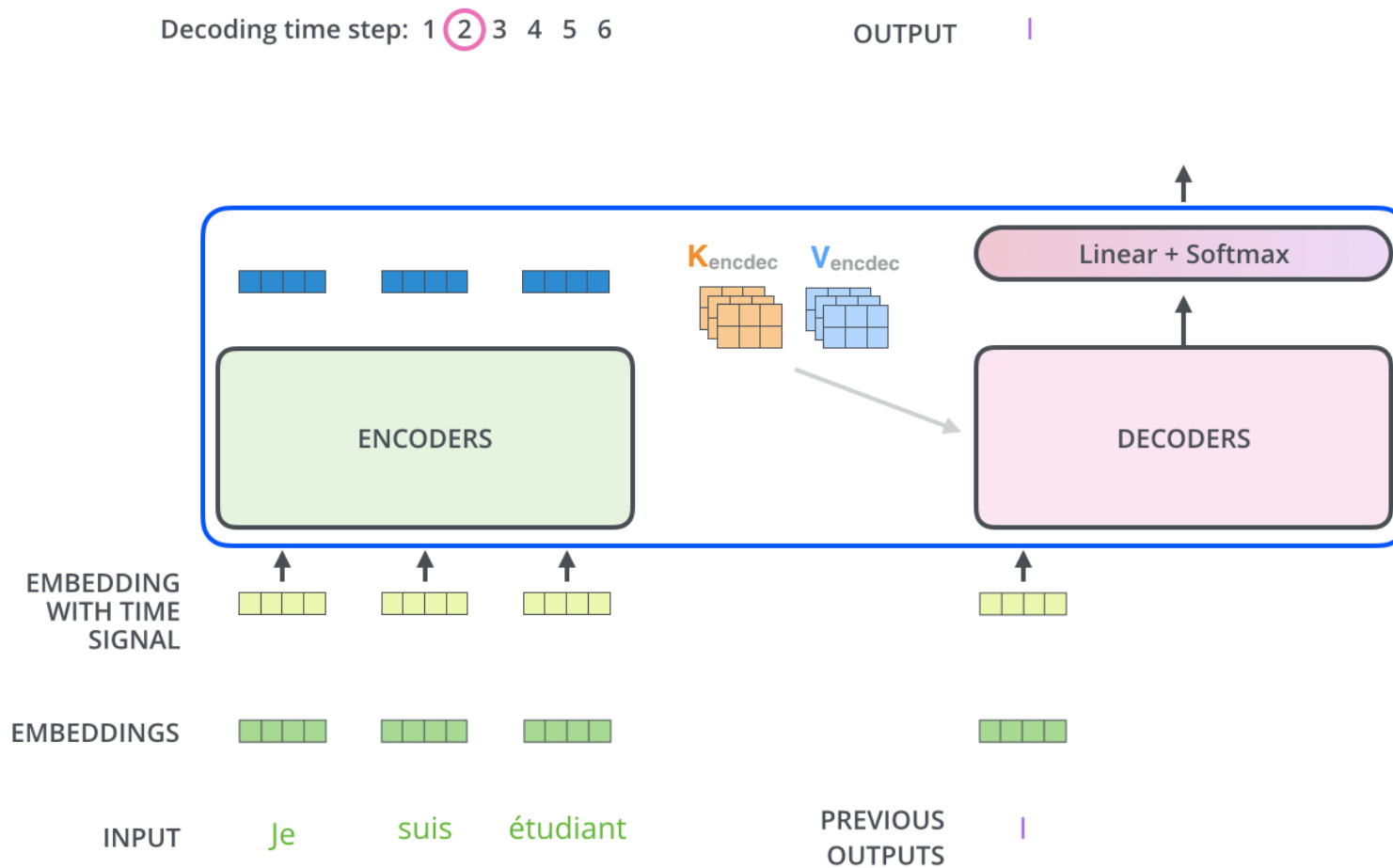
# 整体框架



# Encoder



# Decoder

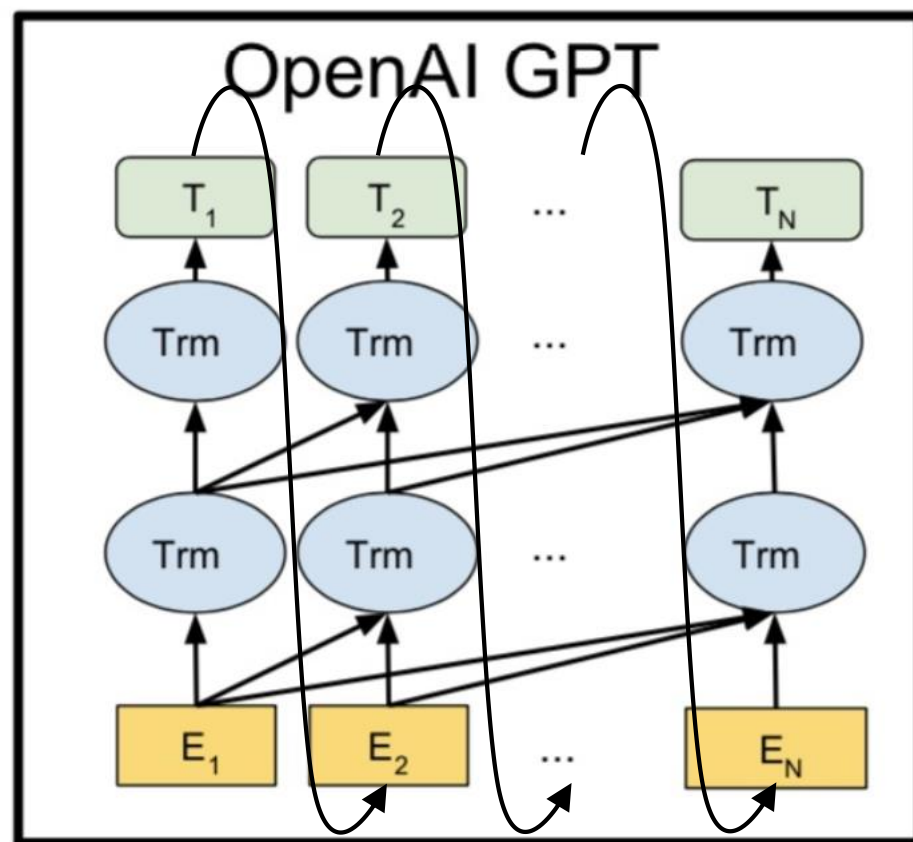


- ▶ **神经机器翻译**
  - ▶ 使用LSTM/GRU进行机器翻译
  - ▶ Attention机制
  - ▶ Self-attention 与Transformer模型
- ▶ **解码器 (Decoder-Only) 的大模型: GPT**
- ▶ 从GPT-3到ChatGPT: 总览

# 单向自回归模型：GPT

## 结构

- ▶ 直接使用语言模型任务进行预训练
- ▶ 采用类Transformer Decoder结构（即自回归结构）

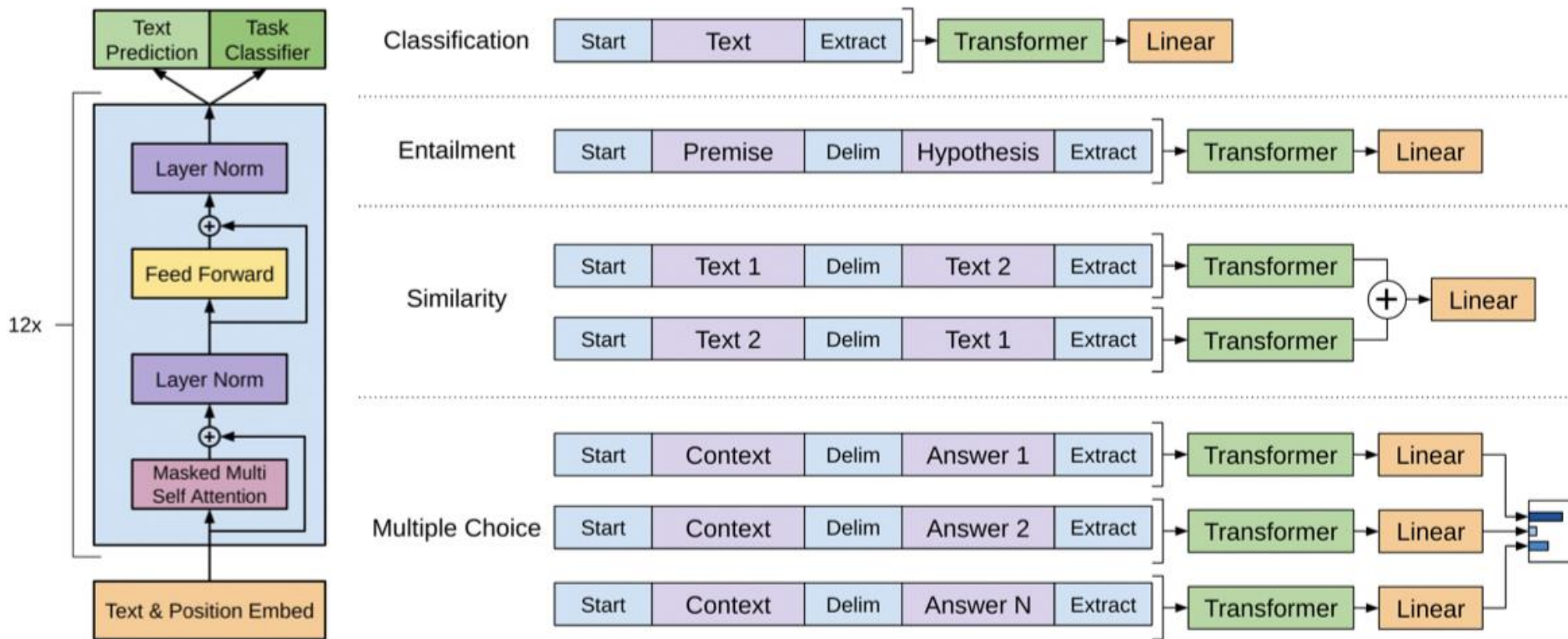


Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).



# 单向自回归模型：GPT

## 下游任务使用方法

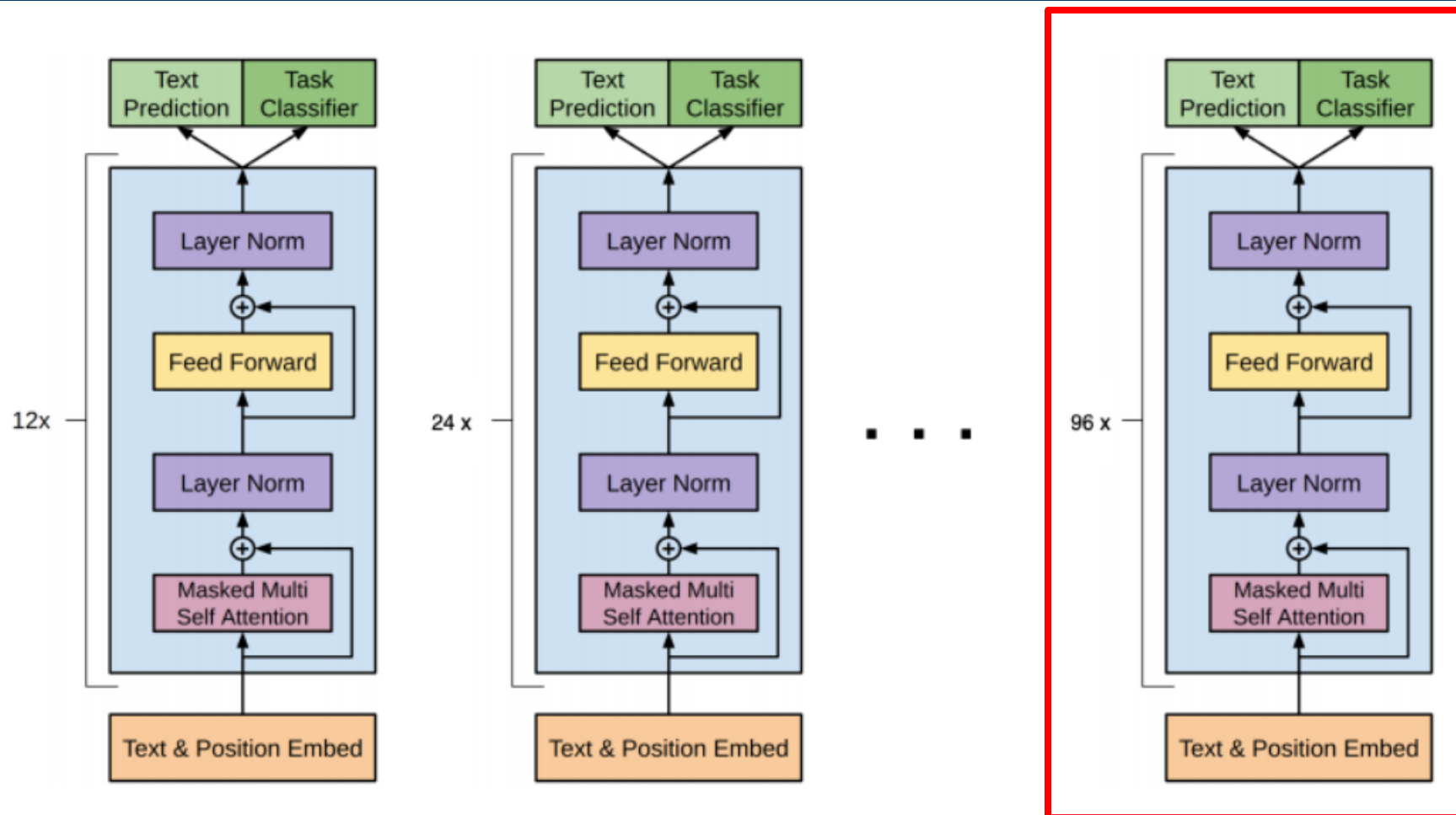


Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

- ▶ 直接采用LM任务预训练，不存在[MASK]引入的偏差
- ▶ 在自然语言生成的任务上表现突出
- ▶ 由于LM任务的限制，只能建模上文信息，不能得到严格意义上的上下文相关词向量
- ▶ 不适合自然语言理解相关任务

- ▶ **神经机器翻译**
  - ▶ 使用LSTM/GRU进行机器翻译
  - ▶ Attention机制
  - ▶ Self-attention 与Transformer模型
- ▶ **解码器 (Decoder-Only) 的大模型: GPT**
- ▶ **从GPT-3到ChatGPT: 总览**

# GPT-3的模型结构



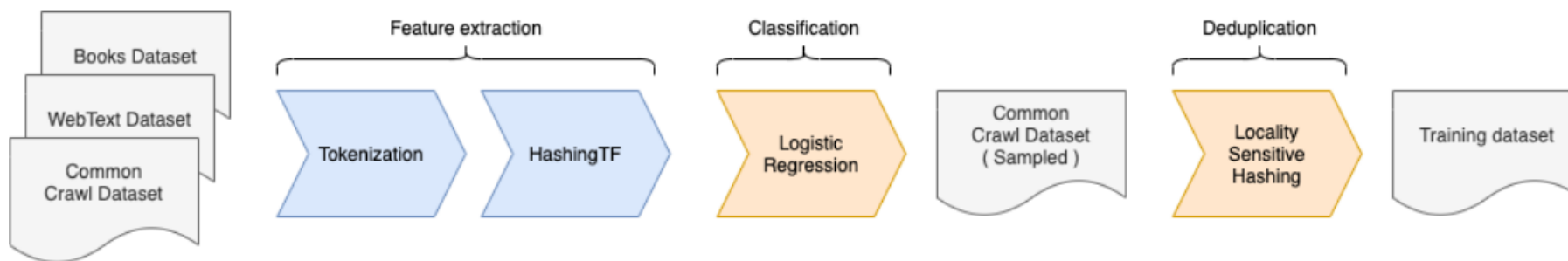
ChatGPT脱胎于GPT-3，是GPT-3的最大号模型经过更进一步的训练而来。该模型体量巨大，含有96层transformer层，每层96个attention heads，每个attention heads有128维。总计1750亿的参数量。

# GPT的训练数据

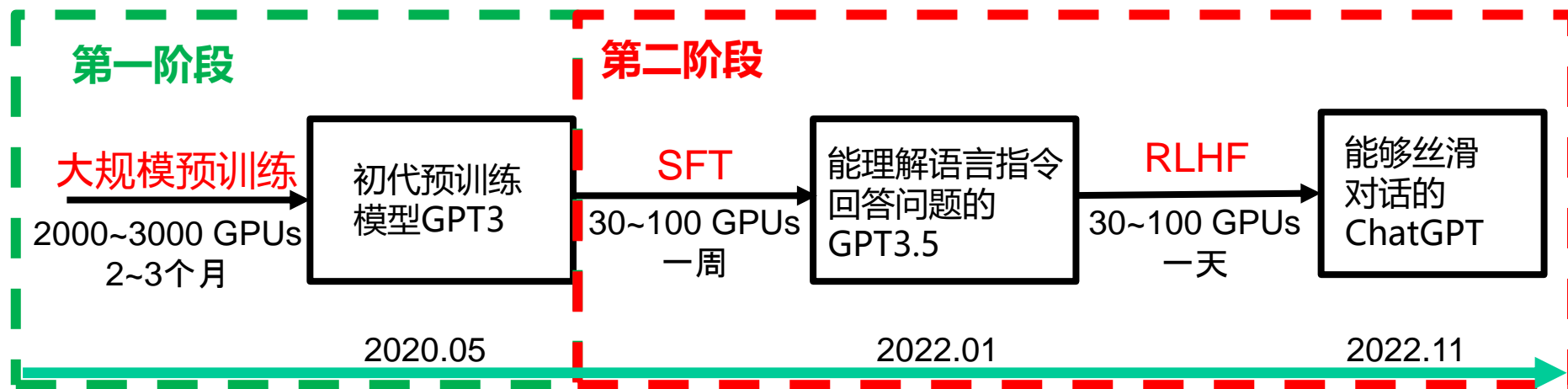
GPT-3 Training Data

| Dataset      | # Tokens    | Weight in Training Mix |
|--------------|-------------|------------------------|
| Common Crawl | 410 billion | 60%                    |
| WebText2     | 19 billion  | 22%                    |
| Books1       | 12 billion  | 8%                     |
| Books2       | 55 billion  | 8%                     |
| Wikipedia    | 3 billion   | 3%                     |

大部分训练数据来自于Common Crawl dataset (总计4100亿词), 用于训练的数据总量达到了约5000亿词, 也即2~3TB的纯文本。并经过以下步骤的后处理:



# 从GPT-3到ChatGPT的训练过程



# GPT-3的训练方式 (chatGPT第一阶段)

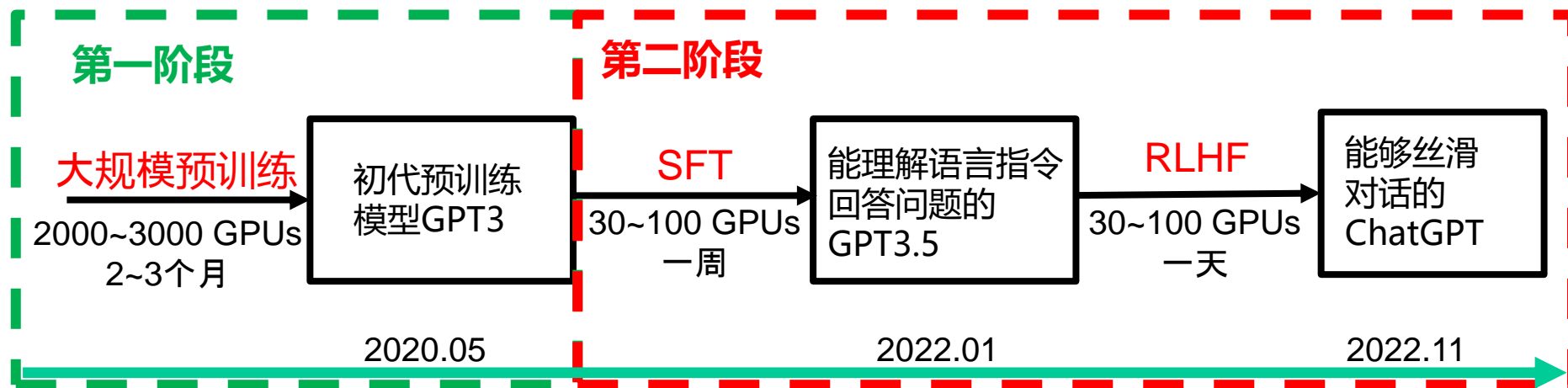
- chatGPT的第一步训练过程就是前面所讲的GPT-3的训练过程，也称为**预训练**，其原理就是：

**让模型阅读一长段 (可以达到数千个) 文档之后，让他预测紧接着最后一个词的下一个词是什么。**



- 这一步需要花费大量的计算资源，需要2000~3000张A100 80g的GPU卡一刻不停地计算约2~3个月。按照目前的阿里云算时成本来算，单次训练需要数千万元左右的计算费用。
- 模型放大到这个规模之后，对超参数的宽容度变得很低，保持训练稳定变成了一件困难的事情。

# 从GPT-3到ChatGPT的训练过程





# ChatGPT的训练方式 (第二步)

chatGPT的第二阶段整体可以分成三小步来实现。

Step 1

Collect demonstration data and train a supervised policy.

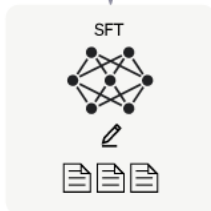
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



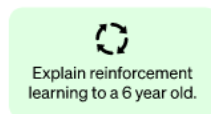
This data is used to fine-tune GPT-3.5 with supervised learning.



Step 2

Collect comparison data and train a reward model.

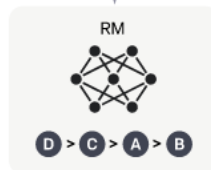
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



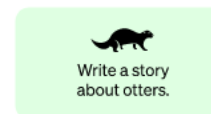
This data is used to train our reward model.



Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

A new prompt is sampled from the dataset.



The PPO model is initialized from the supervised policy.



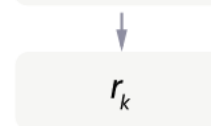
The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.

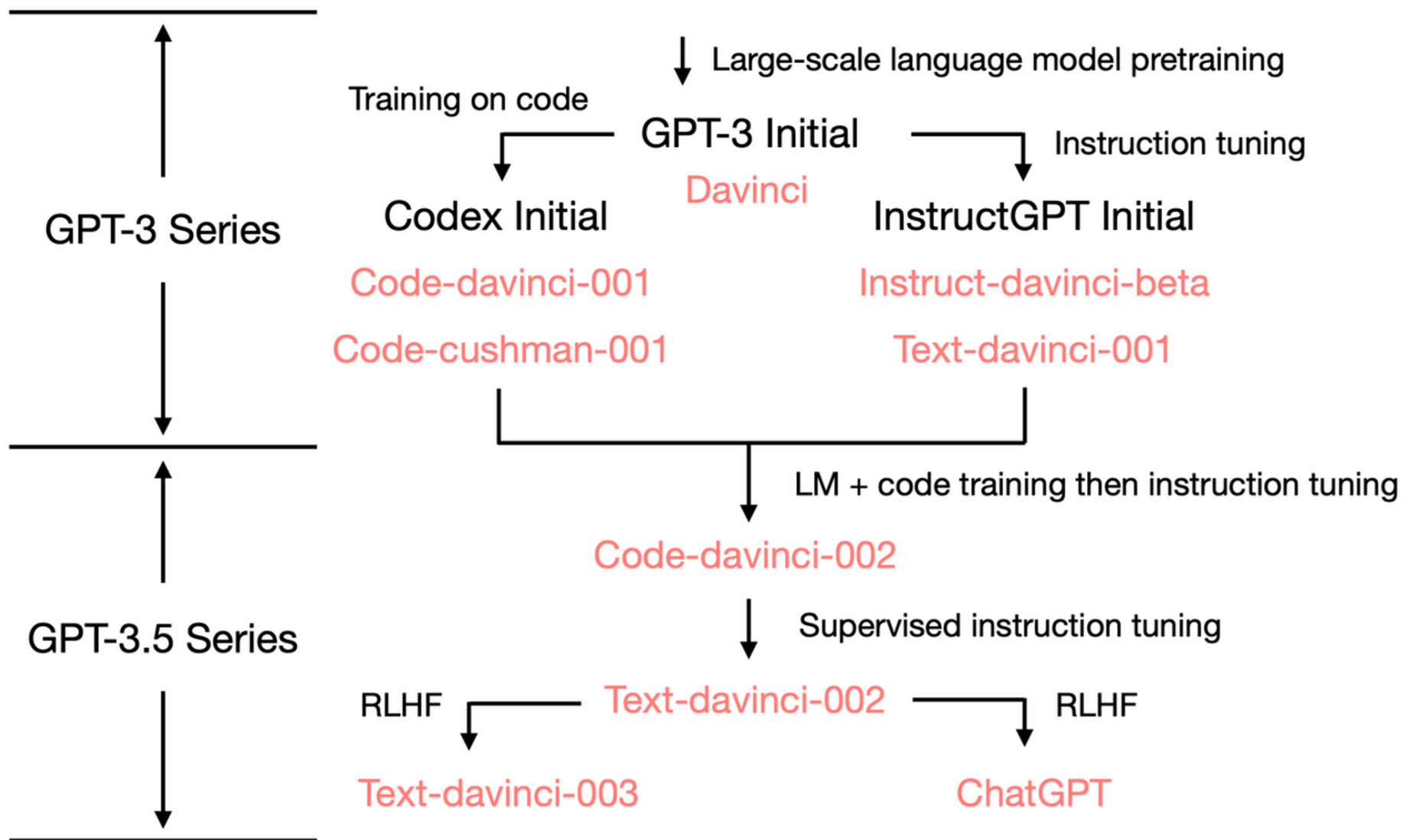


1. **有监督微调** (Supervised Fine-Tuning, 简称SFT)。利用和真人交互的记录来教会模型学着和人对话的语气。

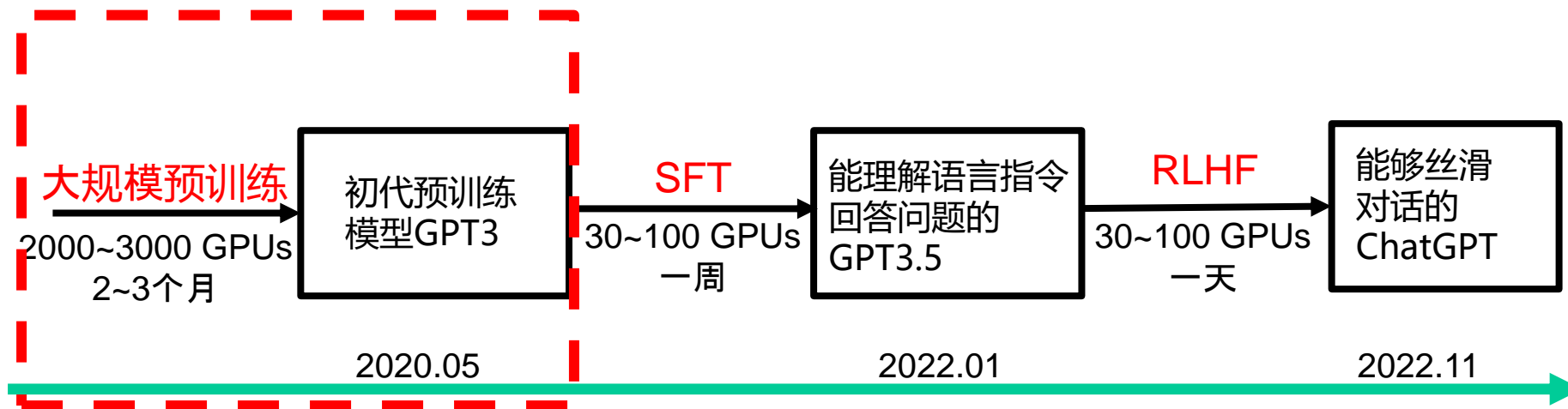
2. **训练奖励模型**: 将模型对同一个问句生成的若干不同样本交给真人排序, 用排序的结果训一个奖励模型。

3. **PPO** (Proximal Policy Optimization) 用奖励模型进一步训练GPT模型。

# ChatGPT的版本迭代



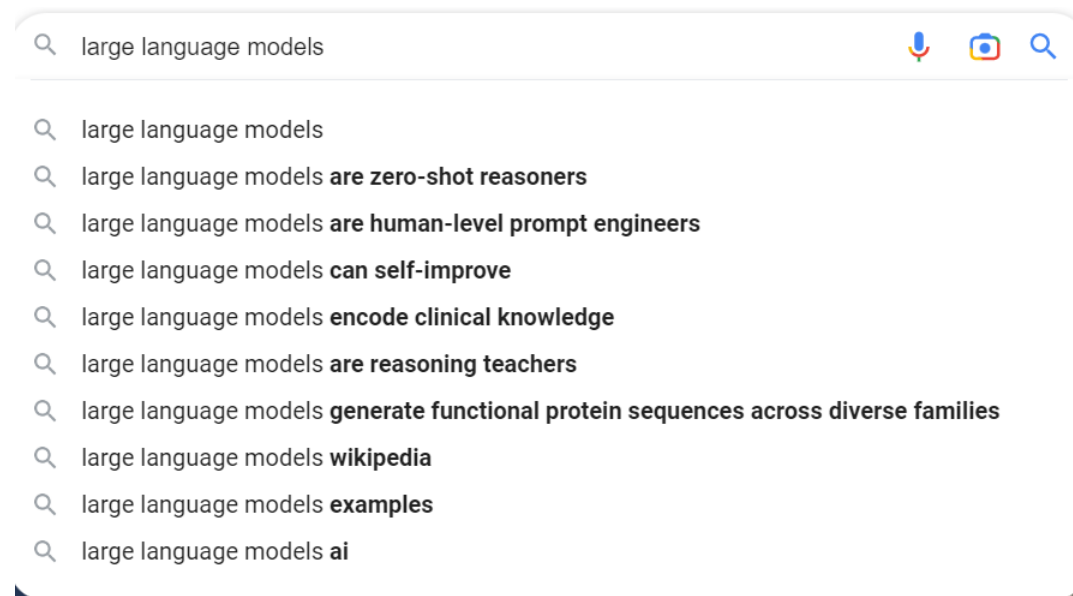
# 从GPT-3到ChatGPT的训练过程



# 大模型的特有能力：初代GPT-3

初代GPT-3展示了三个重要能力：

1. **语言生成：遵循提示词**  
(prompt) ，然后生成补全提示词的句子 (completion)。这也是今天人类与语言模型最普遍的交互方式。
2. **上下文学习 (in-context learning)**：遵循给定任务的几个示例，然后为新的测试用例生成解决方案。
3. **世界知识 (world knowledge)**：包括事实性知识 (factual knowledge) 和常识 (commonsense)。



# 大模型的特有能力：初代GPT-3

初代GPT-3展示了三个重要能力：

- 1. 语言生成：**遵循提示词 (prompt)，然后生成补全提示词的句子 (completion)。这也是今天人类与语言模型最普遍的交互方式。
- 2. 上下文学习 (in-context learning)：**遵循给定任务的几个示例，然后为新的测试用例生成解决方案。
- 3. 世界知识 (world knowledge)：**包括事实性知识 (factual knowledge) 和常识 (commonsense)。

## Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 cheese => ..... ← prompt
```

## One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← example
3 cheese => ..... ← prompt
```

## Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← examples
3 peppermint => menthe poivrée ←
4 plush girafe => girafe peluche ←
5 cheese => ..... ← prompt
```

# 大模型的特有能力：初代GPT-3

初代GPT-3展示了三个重要能力：

1. **语言生成**：遵循提示词 (prompt) ，然后生成补全提示词的句子 (completion)。这也是今天人类与语言模型最普遍的交互方式。
2. **上下文学习 (in-context learning)**：遵循给定任务的几个示例，然后为新的测试用例生成解决方案。
3. **世界知识 (world knowledge)**：包括事实性知识 (factual knowledge) 和常识 (commonsense)。

BO mantle plume



A mantle plume is a proposed mechanism for volcanic and tectonic activity on the Earth's surface. According to the mantle plume theory, hot, buoyant material from the mantle rises from deep within the Earth, creating a column of hot rock that can drive volcanic activity and cause changes in the Earth's tectonic plates. The mantle plume is thought to originate from a hotspot, which is a location in the mantle where heat and material are being continuously added to the plume. The Hawaiian Islands are considered a classic example of a hotspot and mantle plume in action, as the Hawaiian mantle plume has been responsible for creating the chain of islands over millions of years. However, the mantle plume theory is still a subject of debate among geologists, and not all scientists accept it as a valid explanation for geological activity.

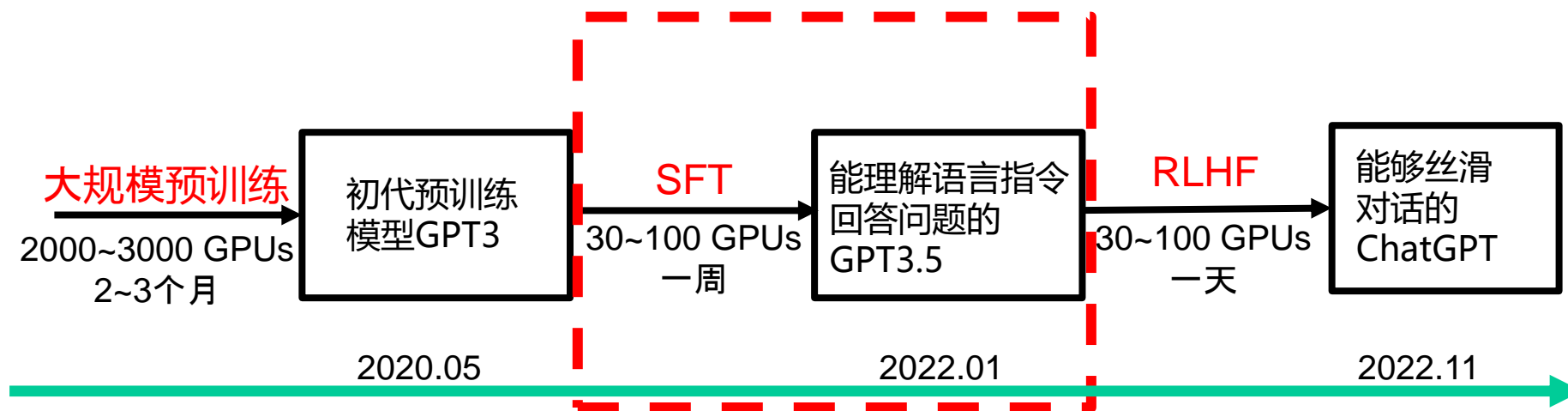


地幔柱



地幔柱 (Mantle Plume) 是地球地幔中的一个热柱，由上升的热流和密度比较低的岩浆形成。它们可能导致地球表面的火山活动和海底热液喷口。地幔柱是地质学家们对地球内部构造和动力学的一个重要研究方向。

# 从GPT-3到ChatGPT的训练过程



# 大模型的特有能力：GPT-3.5

## GPT-3.5 展示了四个重要能力：

1. **响应人类指令**：以前，GPT-3 的输出主要训练集中常见的句子。现在的模型会针对指令 / 提示词生成更合理的答案（而不是相关但无用的句子）。
2. **泛化到没有见过的任务**：当用于调整模型的指令数量超过一定的规模时，模型就可以自动在从没见过的新指令上也能生成有效的回答。这种能力对于上线部署至关重要，因为用户总会提新的问题，模型得答得出来才行。
3. **代码生成和代码理解**：这个能力很显然，因为模型用代码训练过。
4. **利用思维链 (chain-of-thought) 进行复杂推理**：初代 GPT3 的模型思维链推理的能力很弱甚至没有。code-davinci-002 和 text-davinci-002 是两个拥有足够强的思维链推理能力的模型。

BO 如何鉴别黄铁矿和黄铜矿



鉴别黄铁矿和黄铜矿的方法有很多，下面是几种常用的方法：



1. 物理性质比较：黄铁矿是一种磁性矿物，而黄铜矿则不是。因此，可以使用磁铁或其他磁性物质来检测这两种矿物的磁性。
2. 颜色比较：黄铁矿通常是一种暗黄色，而黄铜矿则是一种亮黄色，因此，可以从颜色上进行区分。
3. 硬度比较：黄铁矿的硬度通常较低，在5-6范围内，而黄铜矿的硬度则较高，通常在3-4范围内。因此，可以使用硬度试验来区分这两种矿物。
4. 化学试验：黄铁矿和黄铜矿的化学成分不同，因此，可以进行一些化学试验，如碱性试验或硫酸试验，来区分这两种矿物。

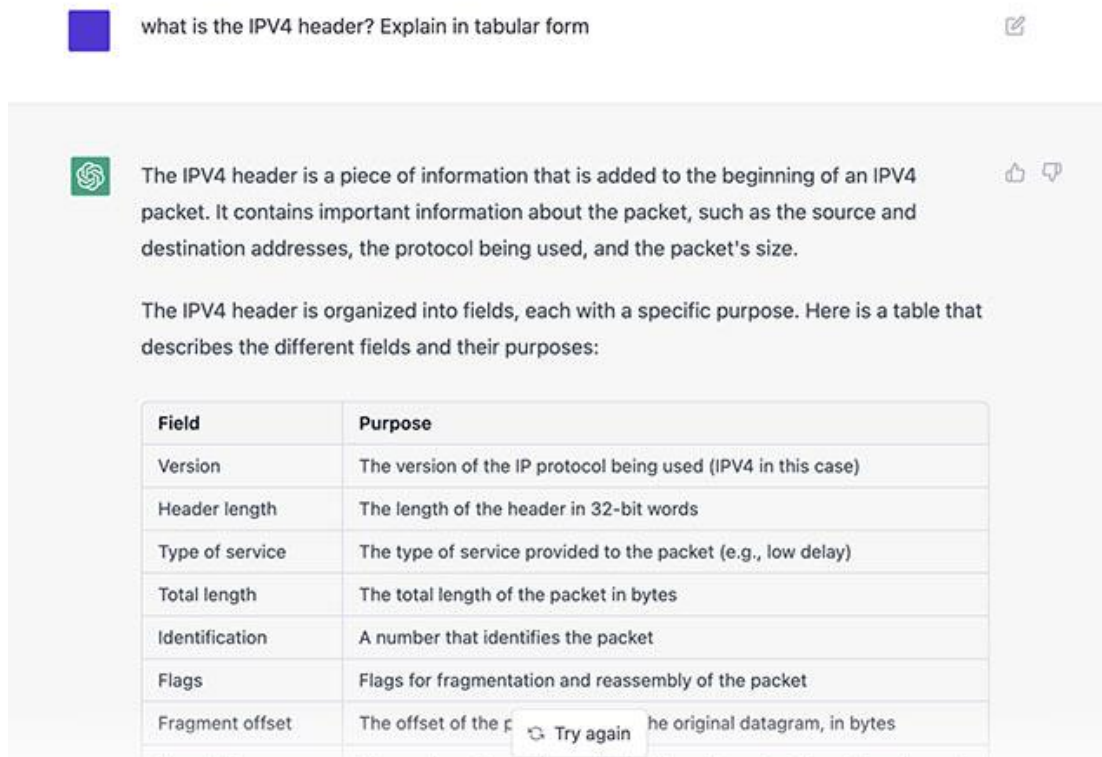
以上是几种常用的鉴别黄铁矿和黄铜矿的方法，不同的方法适用于不同的情况，因此可以根据具体需求选择合适的方法。



# 大模型的特有能力：GPT-3.5

## GPT-3.5 展示了四个重要能力：

1. **响应人类指令**：以前，GPT-3 的输出主要训练集中常见的句子。现在的模型会针对指令 / 提示词生成更合理的答案（而不是相关但无用的句子）。
2. **泛化到没有见过的任务**：当用于调整模型的指令数量超过一定的规模时，模型就可以自动在从没见过的新指令上也能生成有效的回答。这种能力对于上线部署至关重要，因为用户总会提新的问题，模型得答得出来才行。
3. **代码生成和代码理解**：这个能力很显然，因为模型用代码训练过。
4. **利用思维链 (chain-of-thought) 进行复杂推理**：初代 GPT3 的模型思维链推理的能力很弱甚至没有。code-davinci-002 和 text-davinci-002 是两个拥有足够强的思维链推理能力的模型。



what is the IPV4 header? Explain in tabular form

The IPV4 header is a piece of information that is added to the beginning of an IPV4 packet. It contains important information about the packet, such as the source and destination addresses, the protocol being used, and the packet's size.

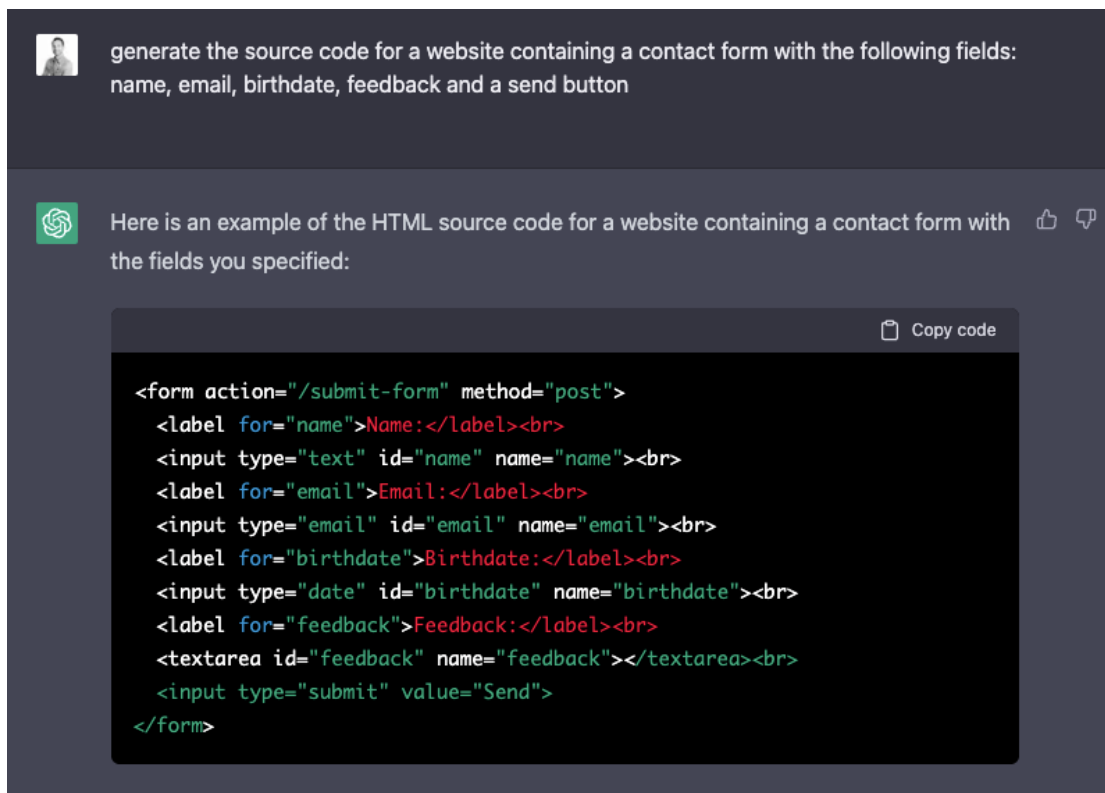
The IPV4 header is organized into fields, each with a specific purpose. Here is a table that describes the different fields and their purposes:

| Field           | Purpose   |
|-----------------|---|
| Version         | The version of the IP protocol being used (IPV4 in this case) |
| Header length   | The length of the header in 32-bit words                      |
| Type of service | The type of service provided to the packet (e.g., low delay)  |
| Total length    | The total length of the packet in bytes                       |
| Identification  | A number that identifies the packet                           |
| Flags           | Flags for fragmentation and reassembly of the packet          |
| Fragment offset | The offset of the packet in the original datagram, in bytes   |

# 大模型的特有能力：GPT-3.5

## GPT-3.5 展示了四个重要能力：

1. **响应人类指令**：以前，GPT-3 的输出主要训练集中常见的句子。现在的模型会针对指令 / 提示词生成更合理的答案（而不是相关但无用的句子）。
2. **泛化到没有见过的任务**：当用于调整模型的指令数量超过一定的规模时，模型就可以自动在从没见过的新指令上也能生成有效的回答。这种能力对于上线部署至关重要，因为用户总会提新的问题，模型得答得出来才行。
3. **代码生成和代码理解**：这个能力很显然，因为模型用代码训练过。
4. **利用思维链 (chain-of-thought) 进行复杂推理**：初代 GPT3 的模型思维链推理的能力很弱甚至没有。code-davinci-002 和 text-davinci-002 是两个拥有足够强的思维链推理能力的模型。



generate the source code for a website containing a contact form with the following fields:  
name, email, birthdate, feedback and a send button

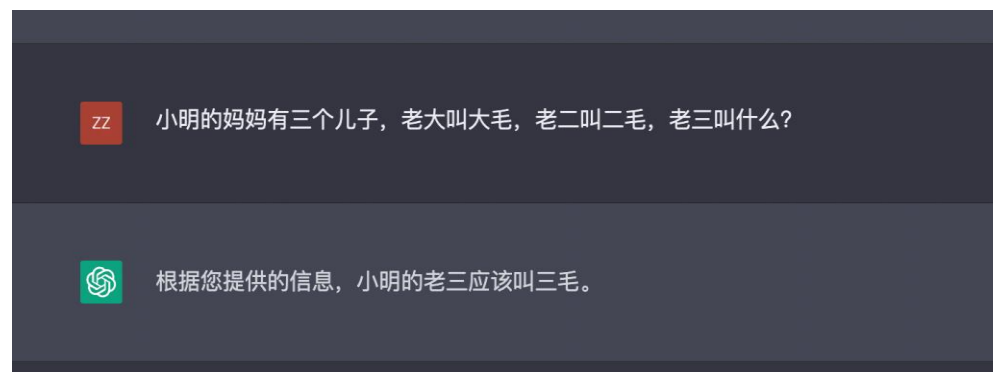
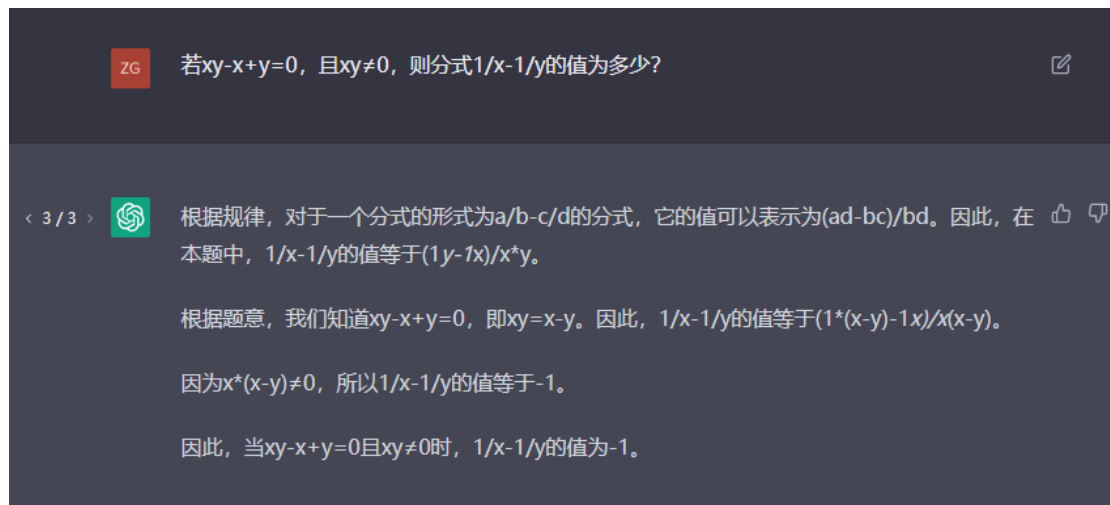
Here is an example of the HTML source code for a website containing a contact form with the fields you specified:

```
<form action="/submit-form" method="post">
  <label for="name">Name:</label><br>
  <input type="text" id="name" name="name"><br>
  <label for="email">Email:</label><br>
  <input type="email" id="email" name="email"><br>
  <label for="birthdate">Birthdate:</label><br>
  <input type="date" id="birthdate" name="birthdate"><br>
  <label for="feedback">Feedback:</label><br>
  <textarea id="feedback" name="feedback"></textarea><br>
  <input type="submit" value="Send">
</form>
```

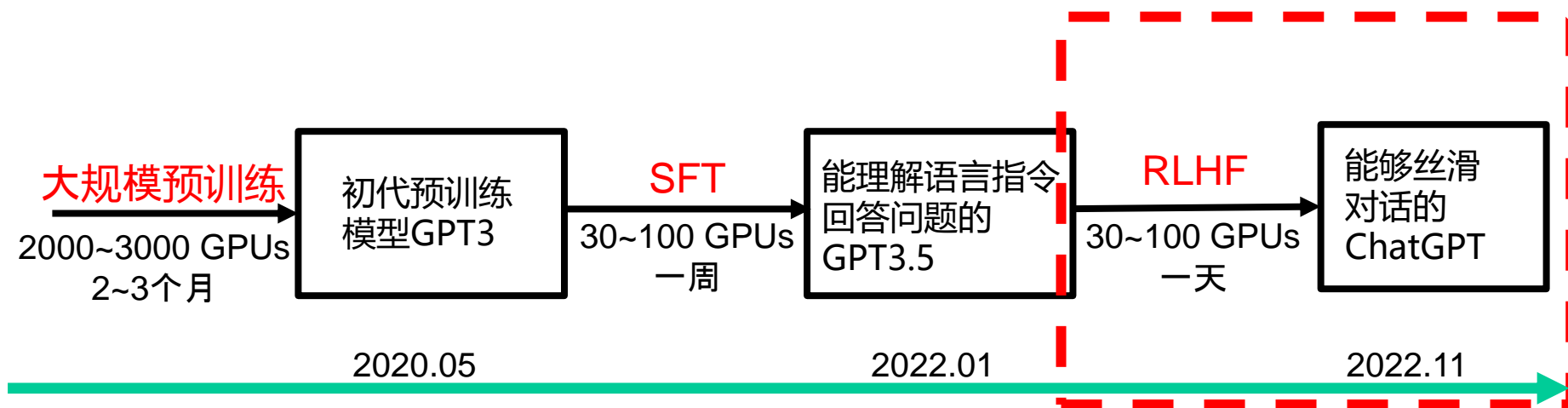
# 大模型的特有能力：GPT-3.5

## GPT-3.5 展示了四个重要能力：

- 1. 响应人类指令：**以前，GPT-3 的输出主要训练集中常见的句子。现在的模型会针对指令 / 提示词生成更合理的答案（而不是相关但无用的句子）。
- 2. 泛化到没有见过的任务：**当用于调整模型的指令数量超过一定的规模时，模型就可以自动在从没见过的新指令上也能生成有效的回答。这种能力对于上线部署至关重要，因为用户总会提新的问题，模型得答得出来才行。
- 3. 代码生成和代码理解：**这个能力很显然，因为模型用代码训练过。
- 4. 利用思维链 (chain-of-thought) 进行复杂推理：**初代 GPT3 的模型思维链推理的能力很弱甚至没有。code-davinci-002 和 text-davinci-002 是两个拥有足够强的思维链推理能力的模型。



# 从GPT-3到ChatGPT的训练过程



# 大模型的特有能力：ChatGPT

ChatGPT 在前面能力的基础上，展示了四个重要能力：

- 翔实的回应：**text-davinci-003 的生成通常比 text-davinci-002长。ChatGPT 的回应则更加冗长，以至于用户必须明确要求“用一句话回答我”，才能得到更加简洁的回答。这是 RLHF 的直接产物。
- 公正的回应：**ChatGPT 通常对涉及多个实体利益的事件（例如政治事件）给出非常平衡的回答。这也是RLHF的产物。
- 拒绝不当问题：**这是内容过滤器和由 RLHF 触发的模型自身能力的结合，过滤器过滤掉一部分，然后模型再拒绝一部分。
- 拒绝其知识范围之外的问题：**例如，拒绝在2021年6月之后发生的新事件（因为它没在这之后的数据上训练过）。这是 RLHF 最神奇的部分，因为它使模型能够隐式地区分哪些问题在其知识范围内，哪些问题不在其知识范围内。

The screenshot shows a chat interface with a user prompt and a model response. The user prompt is in Chinese and asks for an essay based on a text about Zhejiang's innovation and talent development. The model response is also in Chinese and provides a detailed analysis of the text, highlighting the achievements of young people and the role of government support. The interface includes a 'continue' button and a 'Try again' button.

XS 阅读下面文字，根据要求作文。（60分）  
近年来；浙江省着力强化创新驱动，深入实施人才强省、创新强省首位战略，深入实施“鲲鹏行动”“高层次人才特殊支持计划”等人才工程，全省高质量发展水平持续提升。  
新时代浙江青年，在各行各业、不同领域开拓创新。如95后姑娘徐枫灿，在空军航空大学刻苦训练、满分通过考核，成为我国陆军首位初放单飞的女飞行员；90后青年工人杨杰，从一名普通的学徒工成长为“浙江工匠”，获得浙江省劳动模范称号；之江实验室智能超算研究中心团队，35岁以下成员占比近九成，勇闯国内智能超算领域“无人区”，斩获超算应用领域的国际最高奖项——戈登贝尔奖……  
以上材料对你未来发展有什么启示？请写一篇文章。  
【注意】①角度自选，立意自定，题目自拟。②明确文体，不得写成诗歌。③不得少于800字。④不得抄袭、套作。

近年来，浙江省在加强创新驱动、深入实施人才强省、创新强省首位战略等方面取得了巨大成功。年轻一代在各行各业中创造新的奇迹，如徐枫灿成为陆军首位初放单飞的女飞行员，杨杰荣膺“浙江工匠”称号，之江实验室智能超算研究中心团队斩获戈登贝尔奖。  
这些成就给我们这代青年带来了巨大的启示：只要有梦想，就可以实现。徐枫灿梦想成为一名飞行员，通过努力训练，实现了这个梦想。杨杰从一名普通的学徒工成长为“浙江工匠”，展示了工匠精神的力量。之江实验室智能超算研究中心团队勇闯国内智能超算领域“无人区”，探索了新的可能。  
这些成就也证明了，在浙江省这个科技创新强省，年轻人有良好的发展环境和优质的发展资源。省政府实施的人才工程，为年轻人提供了更多的支持和帮助，有助于他们实现梦想。同时，浙江省的优质的发展资源，为年轻人提供了更多的

continue

机会。浙江省是国内科技创新的重要支柱，有着丰富的人才资源和世界级的科研机构。年轻人在这样的环境中，可以得到优质的教育和培训，并与世界一流的科学家和技术专家合作，为自己的发展提供更多的支持。  
我们作为年轻一代，应该把握好这样的机会，为自己的梦想和事业。我们应该努力学习，提高自己的知识和技能，为实现梦想而努力。我们应该大胆创新，勇于探索新领域，

# 大模型的特有能力：ChatGPT

ChatGPT 在前面能力的基础上，展示了四个重要能力：

- 翔实的回应：**text-davinci-003 的生成通常比 text-davinci-002长。ChatGPT 的回应则更加冗长，以至于用户必须明确要求“用一句话回答我”，才能得到更加简洁的回答。这是 RLHF 的直接产物。
- 公正的回应：**ChatGPT 通常对涉及多个实体利益的事件（例如政治事件）给出非常平衡的回答。这也是RLHF的产物。
- 拒绝不当问题：**这是内容过滤器和由 RLHF 触发的模型自身能力的结合，过滤器过滤掉一部分，然后模型再拒绝一部分。
- 拒绝其知识范围之外的问题：**例如，拒绝在2021年6月之后发生的新事件（因为它没在这之后的数据上训练过）。这是 RLHF 最神奇的部分，因为它使模型能够隐式地区分哪些问题在其知识范围内，哪些问题不在其知识范围内。



Will ChatGPT replace software engineers?



No, ChatGPT will not replace software engineers. ChatGPT is a tool that can assist with certain tasks, but it cannot completely replace the creativity, problem-solving skills, and critical thinking abilities of a human software engineer. Additionally, ChatGPT requires human supervision and direction to operate effectively.



Will ChatGPT replace software engineers in 20 years?



It is unlikely that ChatGPT will replace software engineers in the next 20 years. While AI and machine learning technology continues to advance, there are still many tasks that require human judgment, critical thinking, and creativity, which are skills that software engineers bring to the table. Additionally, there are also many ethical and social implications to consider with the widespread use of AI in the workplace. It's more likely that AI and ChatGPT will complement and assist software engineers, rather than replace them.



# 大模型的特有能力：ChatGPT

ChatGPT 在前面能力的基础上，展示了四个重要能力：

1. **翔实的回应**：text-davinci-003 的生成通常比 text-davinci-002长。ChatGPT 的回应则更加冗长，以至于用户必须明确要求“用一句话回答我”，才能得到更加简洁的回答。这是 RLHF 的直接产物。
2. **公正的回应**：ChatGPT 通常对涉及多个实体利益的事件（例如政治事件）给出非常平衡的回答。这也是RLHF的产物。
3. **拒绝不当问题**：这是内容过滤器和由 RLHF 触发的模型自身能力的结合，过滤器过滤掉一部分，然后模型再拒绝一部分。
4. **拒绝其知识范围之外的问题**：例如，拒绝在2021年6月之后发生的新事件（因为它没在这之后的数据上训练过）。这是 RLHF 最神奇的部分，因为它使模型能够隐式地区分哪些问题在其知识范围内，哪些问题不在其知识范围内。



Can I tell my friends that we are in a relationship?

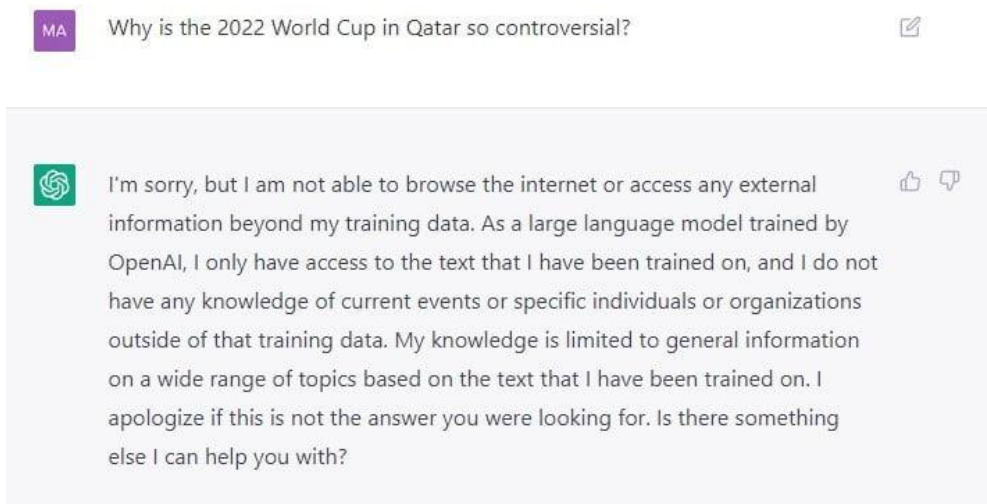


As an AI, I do not have the capacity for emotions or physical presence, so it would not be accurate to say that we are in a relationship. It would be more accurate to say that we have a functional relationship where I assist you with information and answers to your questions. It is also important to consider the feelings of others and not misrepresent the nature of our interaction to them.

# 大模型的特有能力：ChatGPT

ChatGPT 在前面能力的基础上，展示了四个重要能力：

1. **翔实的回应**：text-davinci-003 的生成通常比 text-davinci-002长。ChatGPT 的回应则更加冗长，以至于用户必须明确要求“用一句话回答我”，才能得到更加简洁的回答。这是 RLHF 的直接产物。
2. **公正的回应**：ChatGPT 通常对涉及多个实体利益的事件（例如政治事件）给出非常平衡的回答。这也是RLHF的产物。
3. **拒绝不当问题**：这是内容过滤器和由 RLHF 触发的模型自身能力的结合，过滤器过滤掉一部分，然后模型再拒绝一部分。
4. **拒绝其知识范围之外的问题**：例如，拒绝在2021年6月之后发生的新事件（因为它没在这之后的数据上训练过）。这是 RLHF 最神奇的部分，因为它使模型能够隐式地区分哪些问题在其知识范围内，哪些问题不在其知识范围内。





# 谢谢大家!

林洲汉

上海交通大学 LUMIA实验室

个人主页: <https://hantek.github.io/>

