基于Transformer-XL的MIDI音乐生成

郭成凯

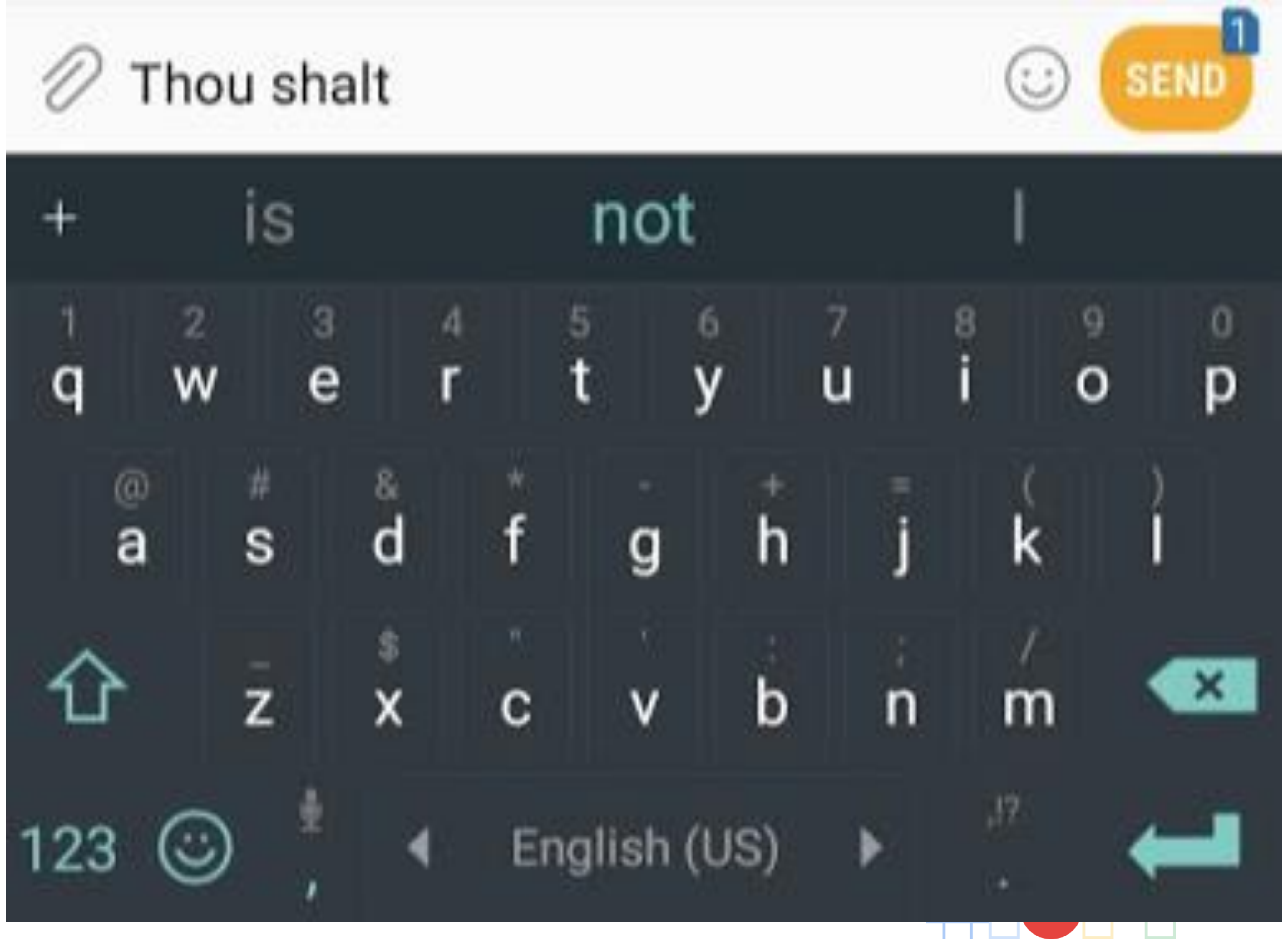
引子

先用Transformer-XL训练小说，效果有局限
后收集了起点总排名前150部小说（自己手动排名...）尝试训练GPT2，失败。。。
尝试midi音乐，先系统调优一下，希望对模型有更深刻的工程实践

GPT2 And Language Modeling

What is a Language Model

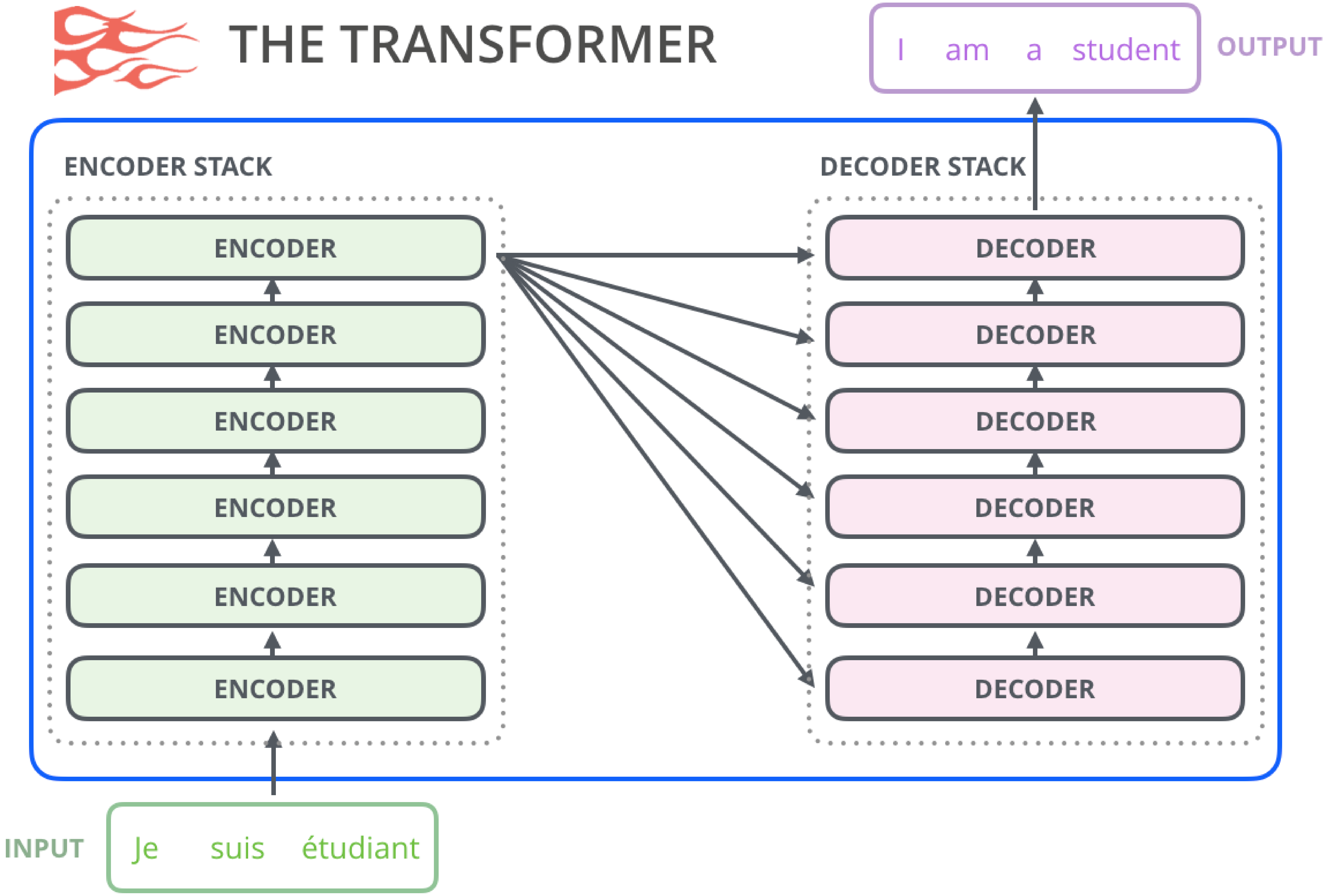
what a language model is – basically a machine learning model that is able to look at part of a sentence and predict the next word. The most famous language models are smartphone keyboards that suggest the next word based on what you've currently typed.



Thou shalt

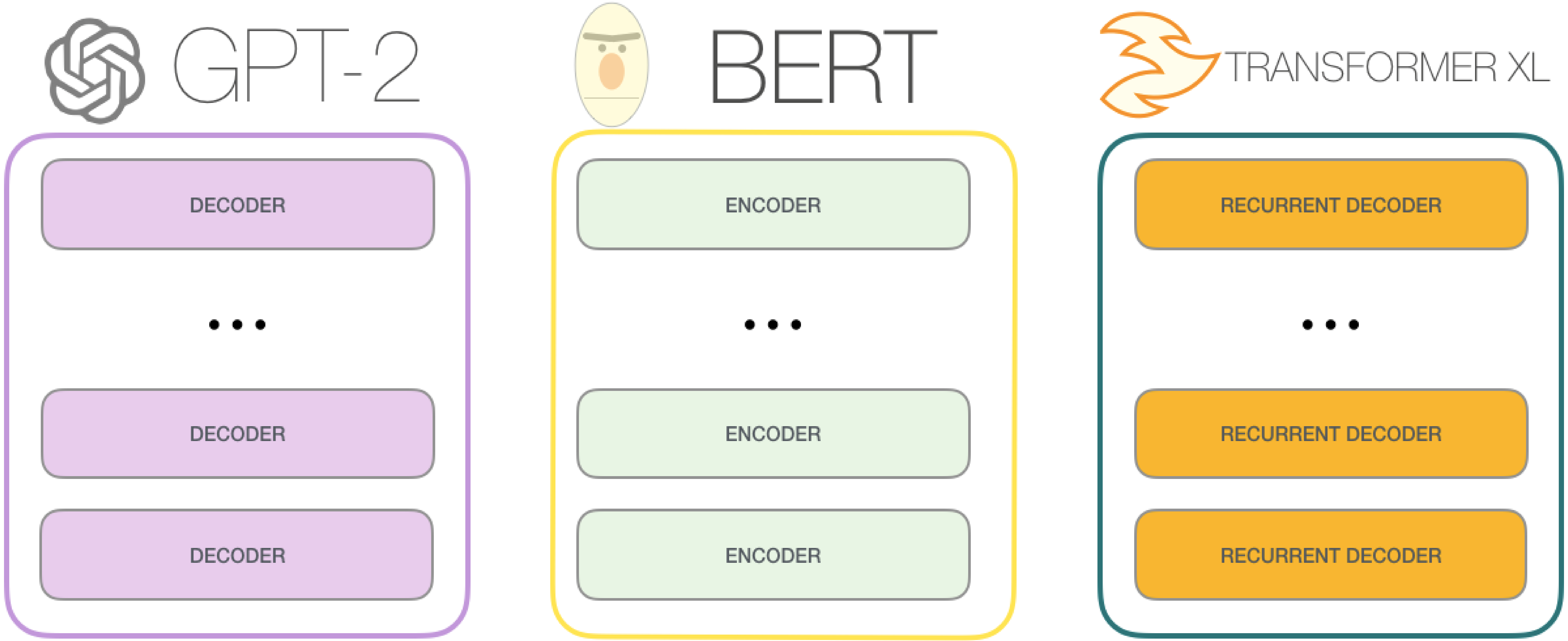
Transformers

- Original Transformer model
- Encoder-Decoder
- Machine Translation



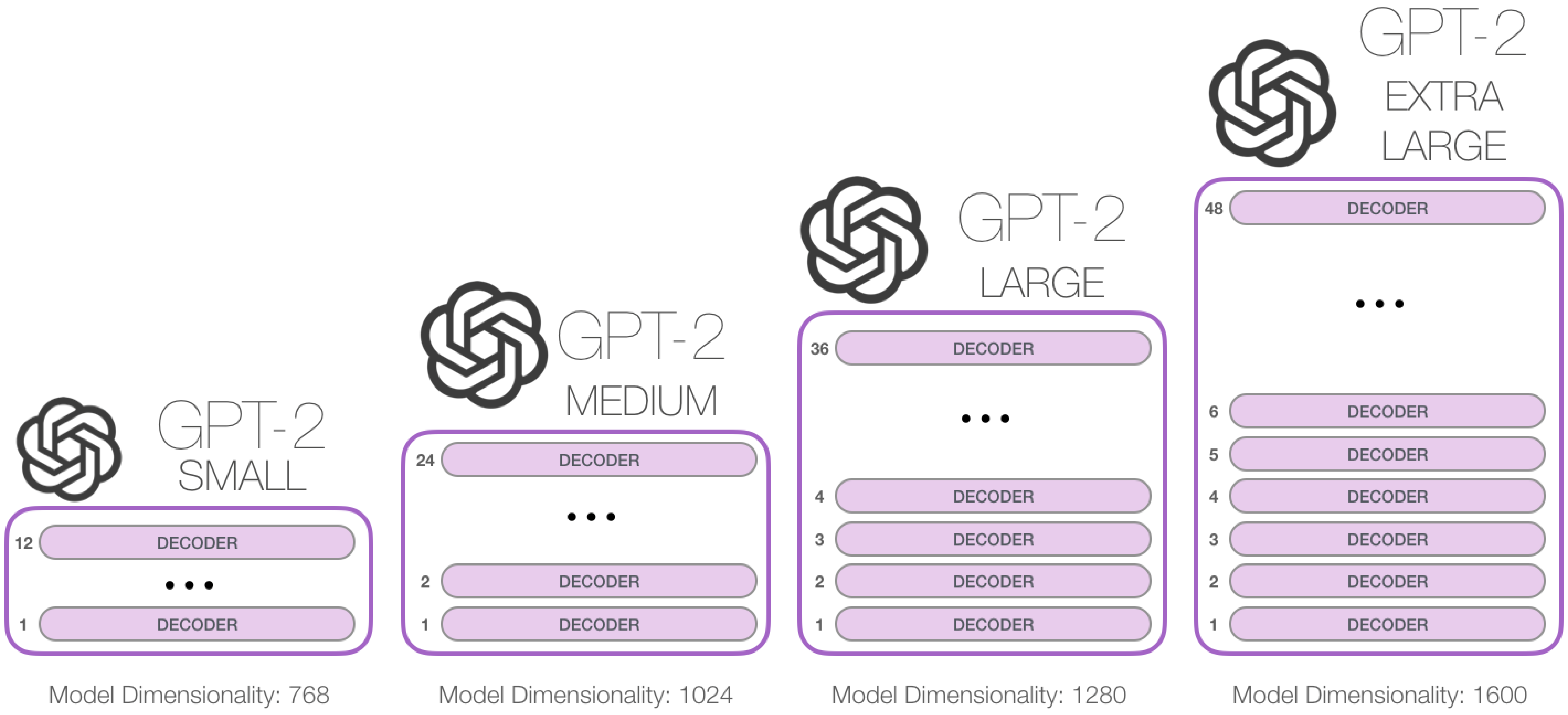
Bert – GPT2 – Transformer-XL

- A lot of the subsequent research work saw the architecture shed either the encoder or decoder, and use just one stack of transformer blocks – stacking them up as high as practically possible, feeding them massive amounts of training text, and throwing vast amounts of compute at them (hundreds of thousands of dollars to train some of these language models, likely millions in the case of AlphaStar).



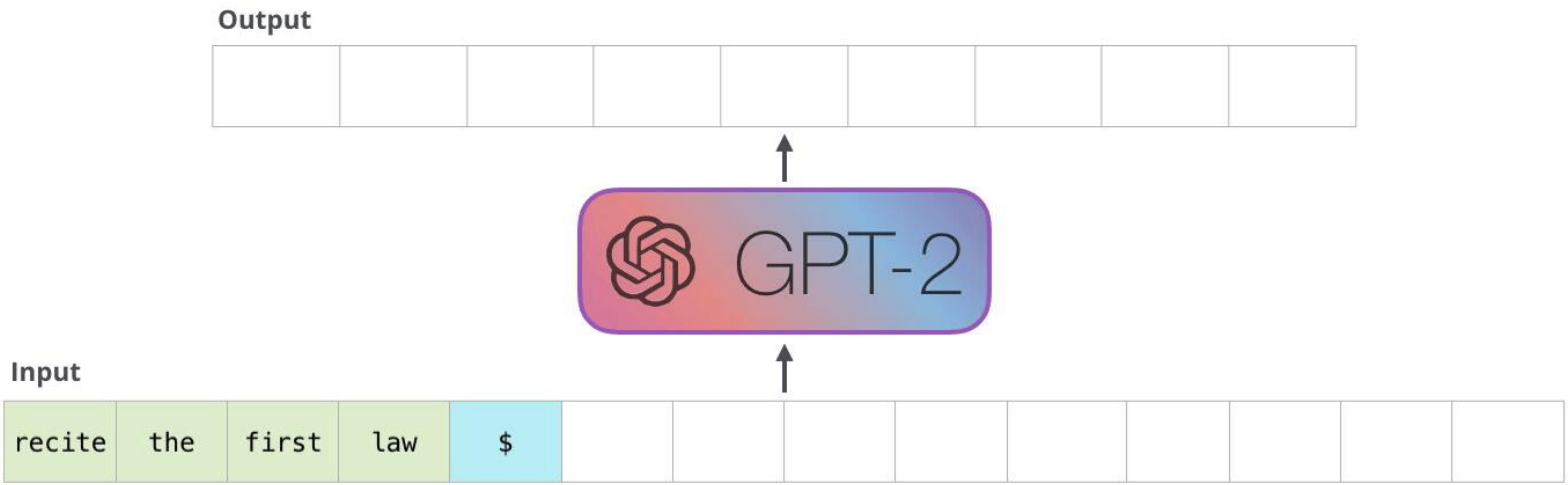
GPT2 Sizes

- 12 layer 768



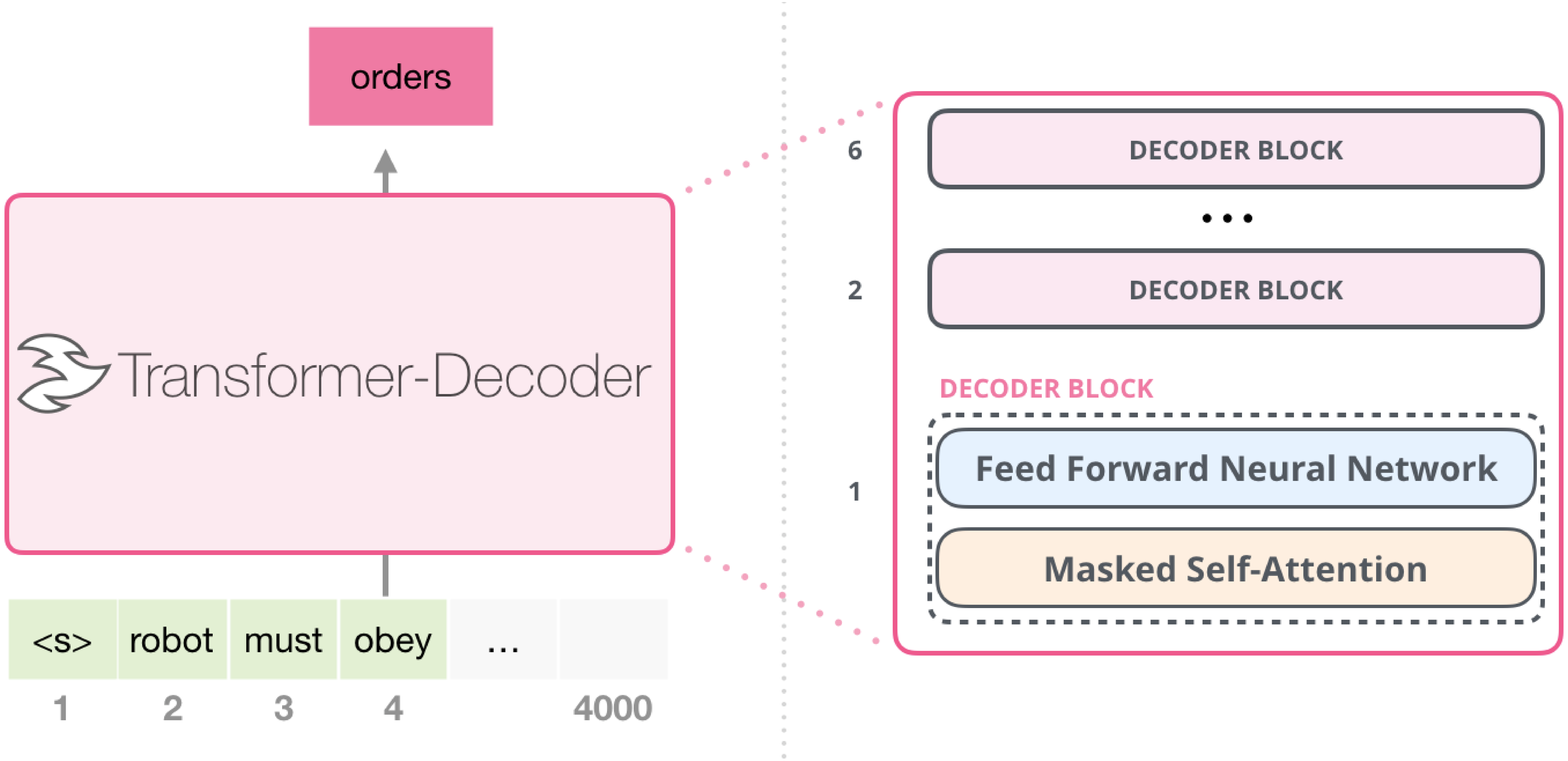
auto-regressive

- AR : after each token is produced, that token is added to the sequence of inputs. And that new sequence becomes the input to the model in its next step
- AE : AE based pretraining does not perform explicit density estimation but instead aims to reconstruct the original data from corrupted input. Bert
- The GPT2, and some later models like TransformerXL and XLNet are auto-regressive in nature. BERT is not. That is a trade off. In losing auto-regression, BERT gained the ability to incorporate the context on both sides of a word to gain better results



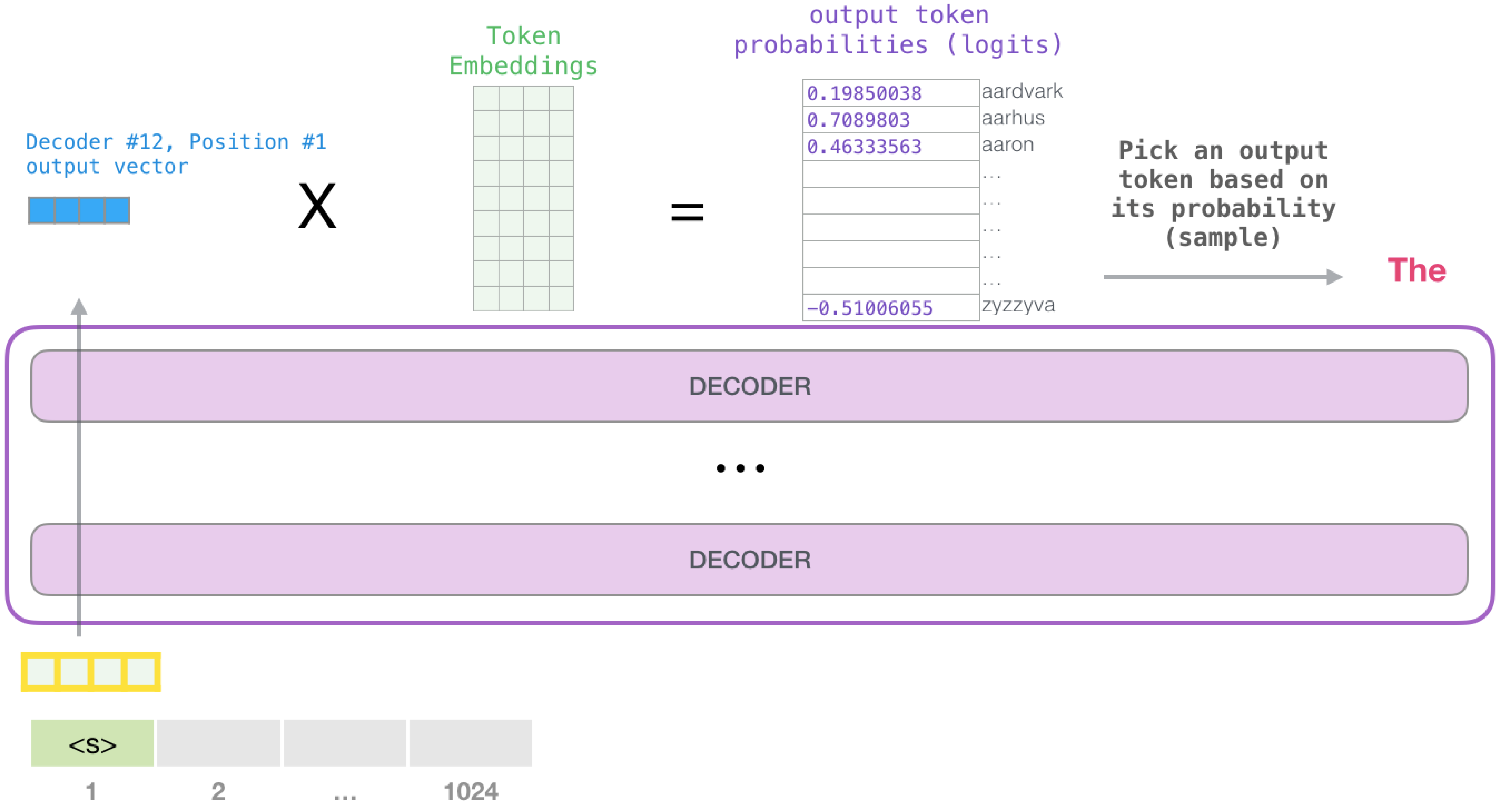
GPT2 Decoder

- masks future tokens



Peek inside

- How GPT2 works?


Token Embeddings
output token probabilities (logits)
Decoder #12, Position #1 output vector
X
=
Pick an output token based on its probability (sample)
The
DECODER
• • •
DECODER
<s>
1
2
...
1024

0.19850038	aardvark
0.7089803	aarhus
0.46333563	aaron
	…
	…
	…
	…
	…
-0.51006055	zyzzyva

Why Transformer-XL

Quoc V. Le https://arxiv.org/search/?searchtype=author&query=Le%2C+Q+V&start=0

AutoML-Zero: Evolving Machine Learning Algorithms From Scratch

Towards a Human-like Open-Domain Chatbot(Meena, a multi-turn open-domain chatbot trained end-to-end on data mined and filtered from public domain social media conversations. This 2.6B parameter neural network is simply trained to minimize perplexity of the next token)

Self-training with Noisy Student improves ImageNet classification

SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition

Listen, Attend and Spell

EfficientDet: Scalable and Efficient Object Detection

EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks

XLNet: Generalized Autoregressive Pretraining for Language Understanding

Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context

Don't Decay the Learning Rate, Increase the Batch Size

Neural Architecture Search with Reinforcement Learning

Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

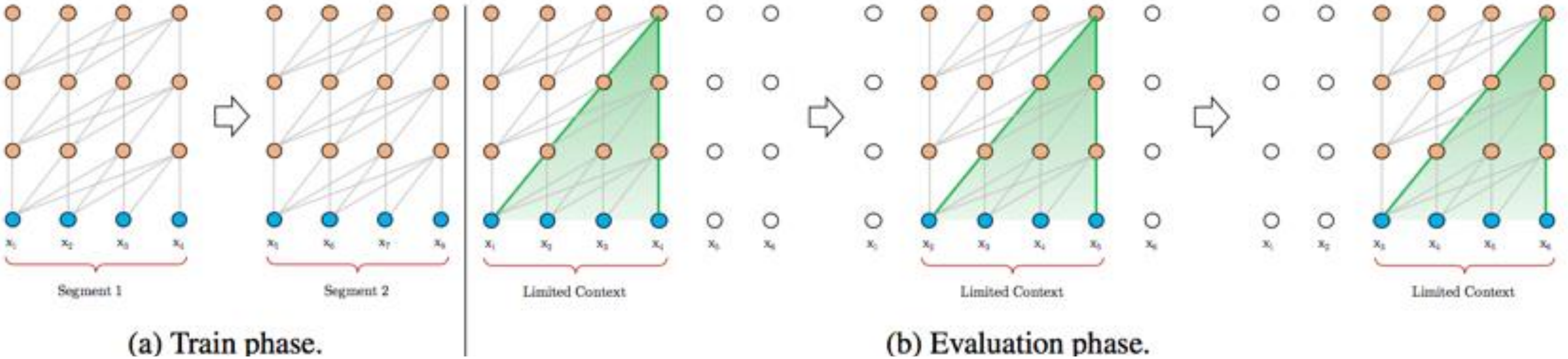
Transformer-XL

- Transformer的限制：Transformer网络具有学习文本更长依赖的能力，但是存在缺陷，transformer模型是在固定长度的segment使用attention学习，因此无法捕获任何超过segment长度的长期依赖性，没有任何跨segment的信息流。且segment的划分是不考虑上下文信息的，会导致上下文碎片化。

- transformer-xl，transformer-xl学习到的依赖比RNN学习到的长80%，比transformer学习到的长450%，transformer-xl两个重要特性
  （1）Segment-level Recurrence with State Reuse
  （2）Relative Position Encodings

Transformer-XL

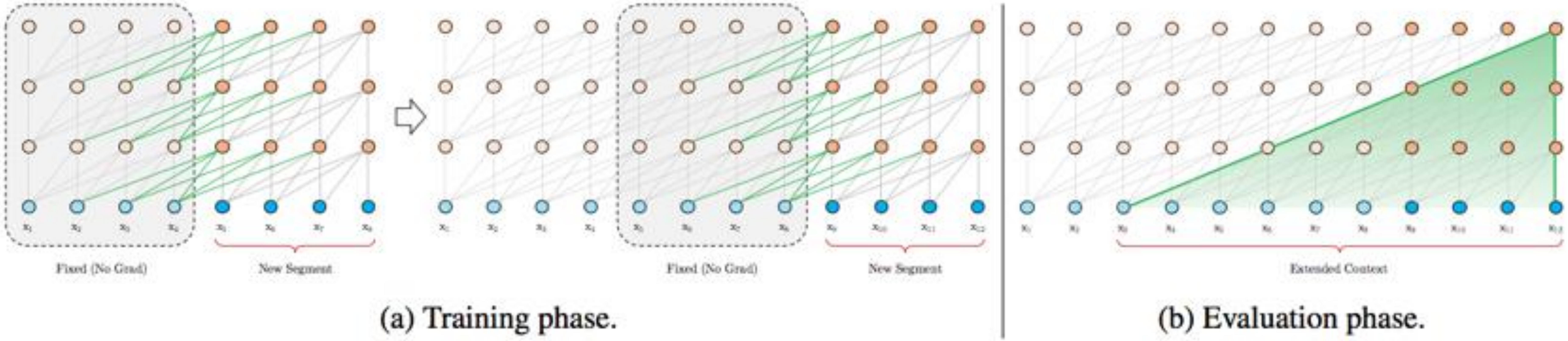
- Segment-Level Recurrence



(a) Train phase.

(b) Evaluation phase.

Transformer



(a) Training phase.

(b) Evaluation phase.

Transformer-XL

Segment-Level Recurrence

$$\begin{aligned}
&\widetilde{\mathbf{h}}_{\tau+1}^{n-1} = \left[\mathrm{SG}(\mathbf{h}_{\tau}^{n-1}) \circ \mathbf{h}_{\tau+1}^{n-1}\right], \\
&\mathbf{q}_{\tau+1}^{n}, \mathbf{k}_{\tau+1}^{n}, \mathbf{v}_{\tau+1}^{n} = \mathbf{h}_{\tau+1}^{n-1}\mathbf{W}_{q}^{\top}, \widetilde{\mathbf{h}}_{\tau+1}^{n-1}\mathbf{W}_{k}^{\top}, \widetilde{\mathbf{h}}_{\tau+1}^{n-1}\mathbf{W}_{v}^{\top}, \\
&\mathbf{h}_{\tau+1}^{n} = \text{Transformer-Layer}\left(\mathbf{q}_{\tau+1}^{n}, \mathbf{k}_{\tau+1}^{n}, \mathbf{v}_{\tau+1}^{n}\right).
\end{aligned}$$

h_{τ}^{n-1}：τ段 n-1 层隐层输出

$h_{\tau+1}^{n-1}$：$\tau+1$段 n-1 层隐层输出

SG()：stop-gradient 函数

$[h_u \circ h_v]$：表示沿着 the length dimension 对$\mathbf{h}_u$和$\mathbf{h}_v$进行 concat

Relative Positional Encoding

$$\begin{aligned}\mathbf{h}_{\tau+1} &= f(\mathbf{h}_{\tau}, \mathbf{E}_{\mathbf{s}_{\tau+1}} + \mathbf{U}_{1:L}) \\ \mathbf{h}_{\tau} &= f(\mathbf{h}_{\tau-1}, \mathbf{E}_{\mathbf{s}_{\tau}} + \mathbf{U}_{1:L}),\end{aligned}$$

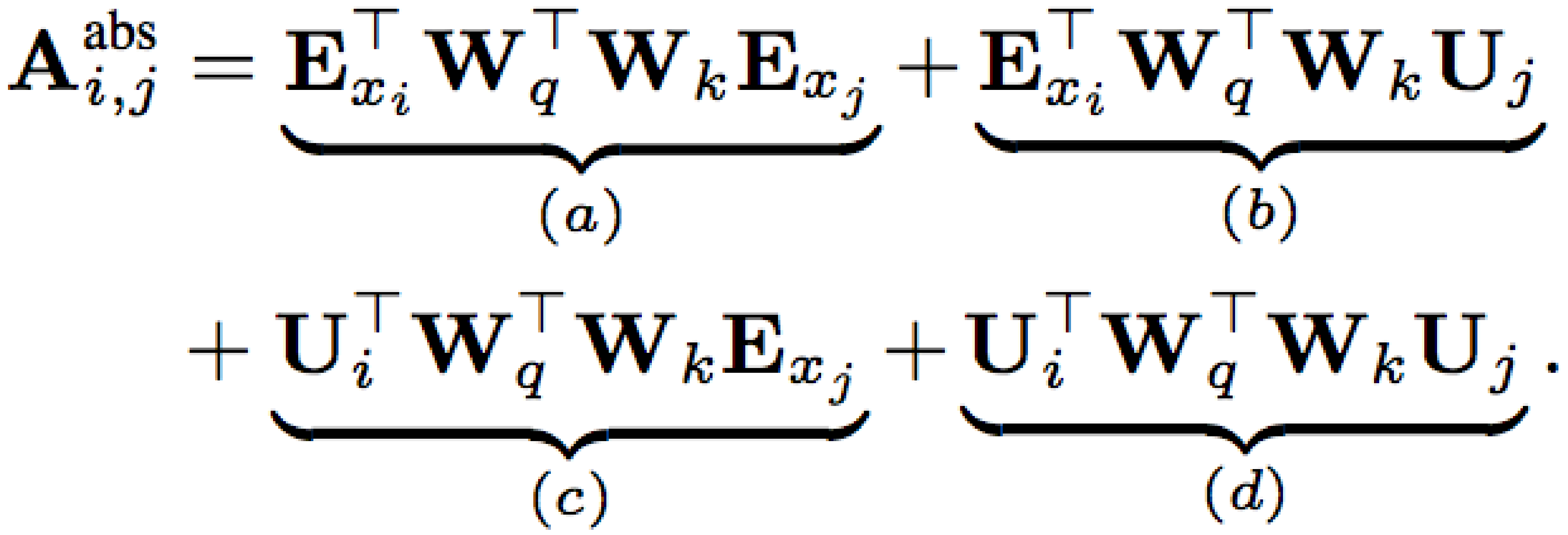
E_{s_τ}： s_τ段的 word embedding

$E_{s_{\tau+1}}$： $s_{\tau+1}$段的 word embedding

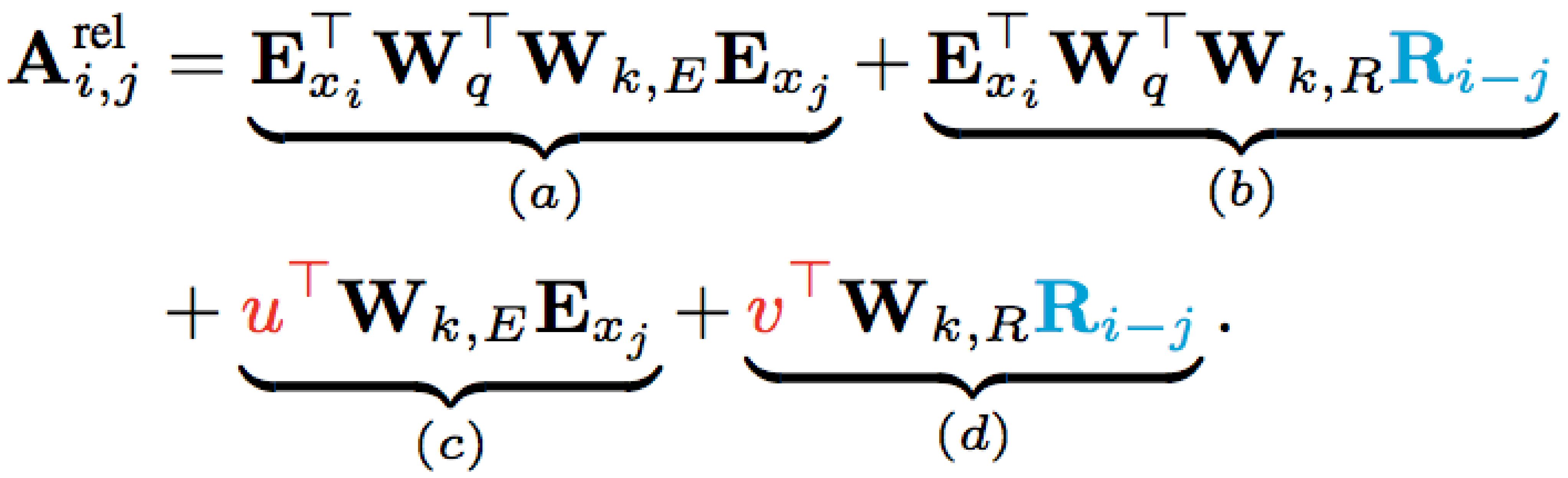
$U_{1:L}$： s_τ段的 position embedding

上面公式展现了如果在segment循环递归机制中仍旧采用绝对位置编码，则s_τ段的信息会传入到$s_{\tau+1}$段中，因此显然s_τ和$s_{\tau+1}$段的位置编码信息就重复了，没有区分度，因此需要采用相对位置编码

Relative Positional Encoding



$$\mathbf{A}^{\text{abs}}_{i,j} = \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{(a)} + \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{U}_j}_{(b)} + \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{(c)} + \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{U}_j}_{(d)}.$$



$$\mathbf{A}^{\text{rel}}_{i,j} = \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_{k,E} \mathbf{E}_{x_j}}_{(a)} + \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_{k,R} \mathbf{R}_{i-j}}_{(b)} + \underbrace{u^\top \mathbf{W}_{k,E} \mathbf{E}_{x_j}}_{(c)} + \underbrace{v^\top \mathbf{W}_{k,R} \mathbf{R}_{i-j}}_{(d)}.$$

$\mathrm{E}_{x_i}^T W_q^T$：x_i的 embedding 信息

$\mathrm{W}_k E_{x_j}$：x_j的 embedding 信息

$\mathrm{U}_i^T W_q^T$：x_i的绝对位置信息

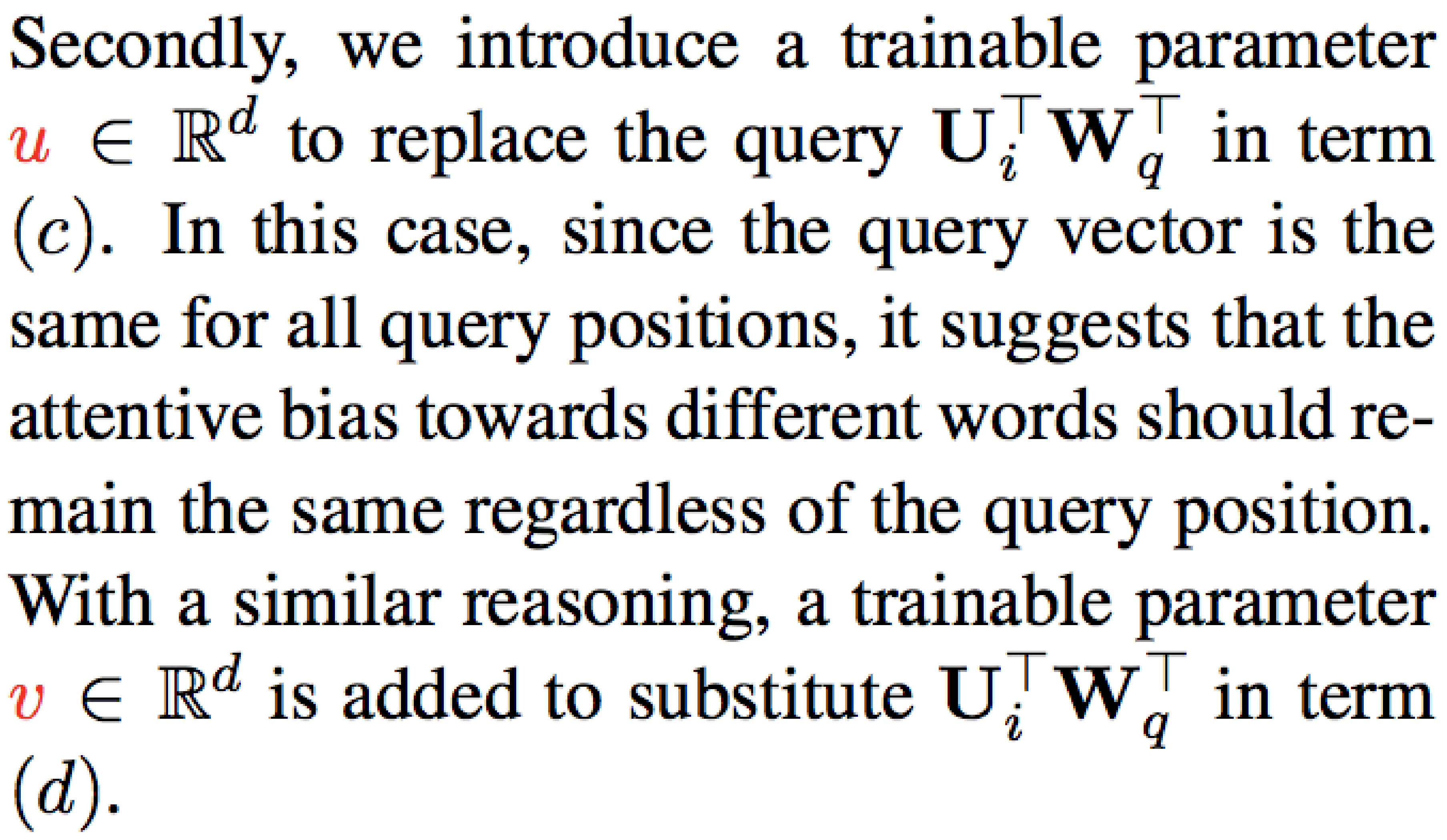
$\mathrm{W}_k U_j$：x_j的绝对位置信息

$\mathrm{E}_{x_i}^T W_q^T$：x_i的 embedding 信息

$\mathrm{W}_{k,\mathrm{E}} E_{x_j}$：$x_j$的 embedding 信息

u^T, v^T代替$\mathrm{U}_i^T W_q^T$：u^T, v^T可训练参数

$\mathrm{W}_{k,R} R_{i-j}$：x_j的相对位置信息



Secondly, we introduce a trainable parameter $u \in \mathbb{R}^d$ to replace the query $\mathbf{U}_i^\top \mathbf{W}_q^\top$ in term (c). In this case, since the query vector is the same for all query positions, it suggests that the attentive bias towards different words should remain the same regardless of the query position. With a similar reasoning, a trainable parameter $v \in \mathbb{R}^d$ is added to substitute $\mathbf{U}_i^\top \mathbf{W}_q^\top$ in term (d).

Transformer-XL

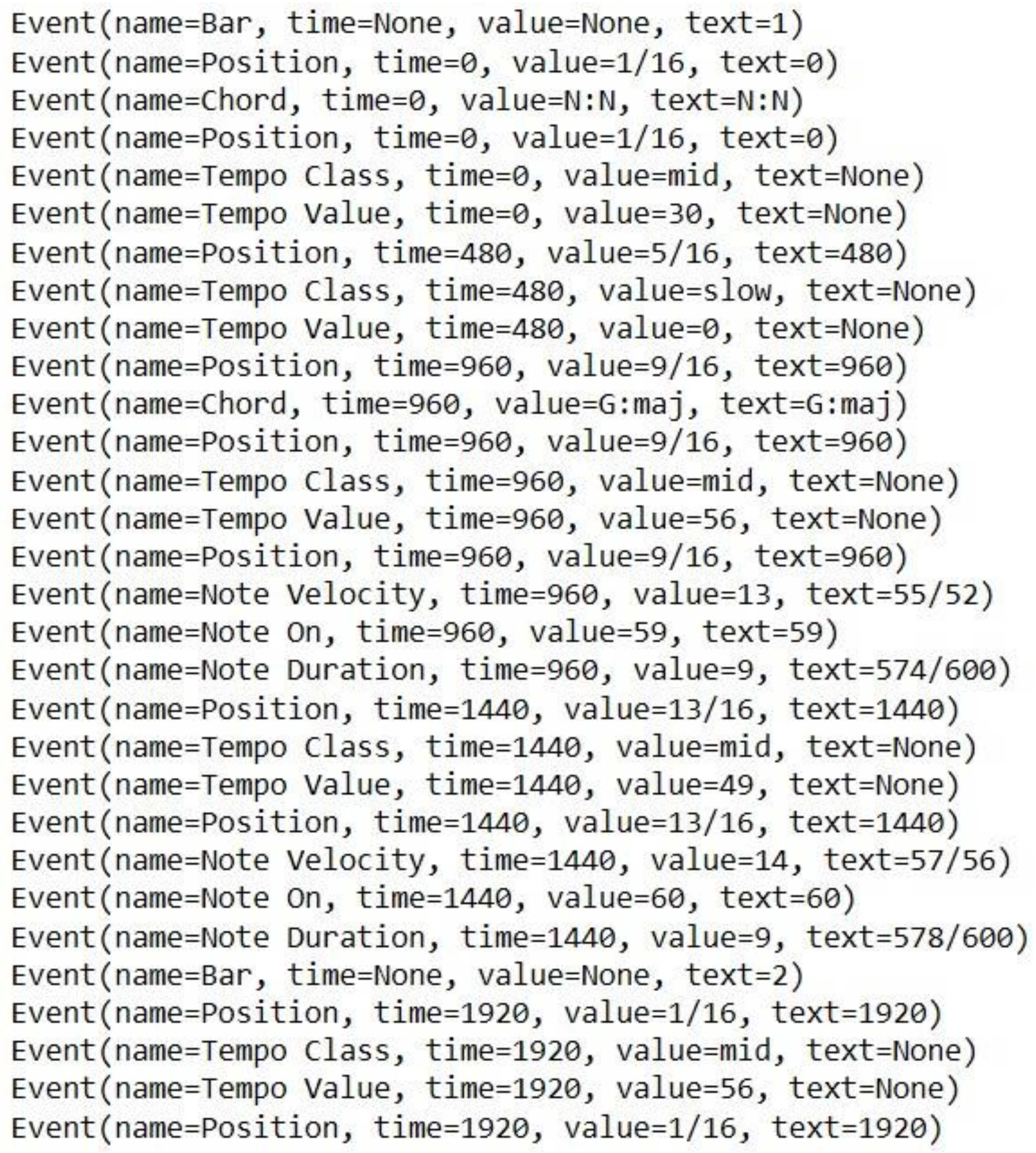
$$\begin{aligned}
\widetilde{\mathbf{h}}_\tau^{n-1} &= \left[\mathrm{SG}(\mathbf{m}_\tau^{n-1}) \circ \mathbf{h}_\tau^{n-1}\right] \\
\mathbf{q}_\tau^n, \mathbf{k}_\tau^n, \mathbf{v}_\tau^n &= \mathbf{h}_\tau^{n-1} {\mathbf{W}_q^n}^\top, \widetilde{\mathbf{h}}_\tau^{n-1} {\mathbf{W}_{k,E}^n}^\top, \widetilde{\mathbf{h}}_\tau^{n-1} {\mathbf{W}_v^n}^\top \\
\mathbf{A}_{\tau,i,j}^n &= {\mathbf{q}_{\tau,i}^n}^\top \mathbf{k}_{\tau,j}^n + {\mathbf{q}_{\tau,i}^n}^\top \mathbf{W}_{k,R}^n \mathbf{R}_{i-j} \\
&\quad + u^\top \mathbf{k}_{\tau,j} + v^\top \mathbf{W}_{k,R}^n \mathbf{R}_{i-j} \\
\mathbf{a}_\tau^n &= \text{Masked-Softmax}(\mathbf{A}_\tau^n) \mathbf{v}_\tau^n \\
\mathbf{o}_\tau^n &= \text{LayerNorm}(\text{Linear}(\mathbf{a}_\tau^n) + \mathbf{h}_\tau^{n-1}) \\
\mathbf{h}_\tau^n &= \text{Positionwise-Feed-Forward}(\mathbf{o}_\tau^n)
\end{aligned}$$

$$h_{z_t}^{(m)} \leftarrow \text{Attention}(\mathrm{Q} = h_{z_t}^{(m-1)}, \mathrm{KV} = \left[\tilde{\mathbf{h}}^{(m-1)}, \mathbf{h}_{\mathbf{z}_{\leq t}}^{(m-1)}\right]; \theta)$$

MIDI音乐生成

NLP for Music

- midi音乐文件可以处理为如下序列



```
Event(name=Bar, time=None, value=None, text=1)
Event(name=Position, time=0, value=1/16, text=0)
Event(name=Chord, time=0, value=N:N, text=N:N)
Event(name=Position, time=0, value=1/16, text=0)
Event(name=Tempo Class, time=0, value=mid, text=None)
Event(name=Tempo Value, time=0, value=30, text=None)
Event(name=Position, time=480, value=5/16, text=480)
Event(name=Tempo Class, time=480, value=slow, text=None)
Event(name=Tempo Value, time=480, value=0, text=None)
Event(name=Position, time=960, value=9/16, text=960)
Event(name=Chord, time=960, value=G:maj, text=G:maj)
Event(name=Position, time=960, value=9/16, text=960)
Event(name=Tempo Class, time=960, value=mid, text=None)
Event(name=Tempo Value, time=960, value=56, text=None)
Event(name=Position, time=960, value=9/16, text=960)
Event(name=Note Velocity, time=960, value=13, text=55/52)
Event(name=Note On, time=960, value=59, text=59)
Event(name=Note Duration, time=960, value=9, text=574/600)
Event(name=Position, time=1440, value=13/16, text=1440)
Event(name=Tempo Class, time=1440, value=mid, text=None)
Event(name=Tempo Value, time=1440, value=49, text=None)
Event(name=Position, time=1440, value=13/16, text=1440)
Event(name=Note Velocity, time=1440, value=14, text=57/56)
Event(name=Note On, time=1440, value=60, text=60)
Event(name=Note Duration, time=1440, value=9, text=578/600)
Event(name=Bar, time=None, value=None, text=2)
Event(name=Position, time=1920, value=1/16, text=1920)
Event(name=Tempo Class, time=1920, value=mid, text=None)
Event(name=Tempo Value, time=1920, value=56, text=None)
Event(name=Position, time=1920, value=1/16, text=1920)
```

NLP for Music

- 字典
- 共320类

```
{'Bar_None' : 0, 'Position_1/16' : 1, 'Tempo Class_mid' : 2, 'Tempo Value_30' : 3, 'Position_5/16' : 4, 'Position_9/16' : 5, 'Position_13/16' : 6, 'Tempo Class_slow' : 7, 'Tempo Value_33' : 8, 'Posi
tion_16/16' : 9, 'Note Velocity_12' : 10, 'Note On_72' : 11, 'Note Duration_1' : 12, 'Tempo Value_25' : 13, 'Note Velocity_14' : 14, 'Note On_76' : 15, 'Note Duration_16' : 16, 'Note Velocity_16' :
17, 'Note On_79' : 18, 'Note On_60' : 19, 'Note On_64' : 20, 'Note Velocity_13' : 21, 'Note On_67' : 22, 'Note On_71' : 23, 'Note On_83' : 24, 'Tempo Value_17' : 25, 'Note Velocity_18' : 26, 'Note O
n_86' : 27, 'Note Duration_8' : 28, 'Position_8/16' : 29, 'Note On_74' : 30, 'Note On_78' : 31, 'Note Velocity_11' : 32, 'Note On_59' : 33, 'Note Velocity_9' : 34, 'Note On_62' : 35, 'Note On_66' : 3
6, 'Note On_69' : 37, 'Note Duration_13' : 38, 'Note Velocity_17' : 39, 'Note On_81' : 40, 'Note Duration_32' : 41, 'Note On_57' : 42, 'Note Duration_15' : 43, 'Note Duration_31' : 44, 'Tempo Value
_27' : 45, 'Tempo Value_21' : 46, 'Tempo Value_19' : 47, 'Note Velocity_15' : 48, 'Note Duration_3' : 49, 'Note Duration_2' : 50, 'Tempo Value_23' : 51, 'Note Duration_17' : 52, 'Note Velocity_10' :
 53, 'Note Duration_19' : 54, 'Note Duration_30' : 55, 'Position_15/16' : 56, 'Note Duration_0' : 57, 'Note Velocity_19' : 58, 'Tempo Value_35' : 59, 'Note Duration_33' : 60, 'Note Duration_23' : 6
1, 'Note Duration_11' : 62, 'Note On_84' : 63, 'Position_3/16' : 64, 'Note Duration_27' : 65, 'Tempo Value_32' : 66, 'Note Duration_24' : 67, 'Position_7/16' : 68, 'Tempo Value_13' : 69, 'Note Dura
tion_34' : 70, 'Note Duration_9' : 71, 'Note Duration_21' : 72, 'Position_11/16' : 73, 'Note Duration_12' : 74, 'Note Duration_6' : 75, 'Position_14/16' : 76, 'Note Duration_5' : 77, 'Note On_48' :
78, 'Note On_36' : 79, 'Note On_35' : 80, 'Note On_47' : 81, 'Note On_33' : 82, 'Note Duration_20' : 83, 'Note On_45' : 84, 'Note On_52' : 85, 'Note On_38' : 86, 'Note On_50' : 87, 'Note On_40' : 88,
 'Note Duration_22' : 89, 'Note Duration_7' : 90, 'Tempo Value_40' : 91, 'Note Duration_18' : 92, 'Note On_55' : 93, 'Note On_43' : 94, 'Note On_42' : 95, 'Note On_54' : 96, 'Note Duration_4' : 97,
```

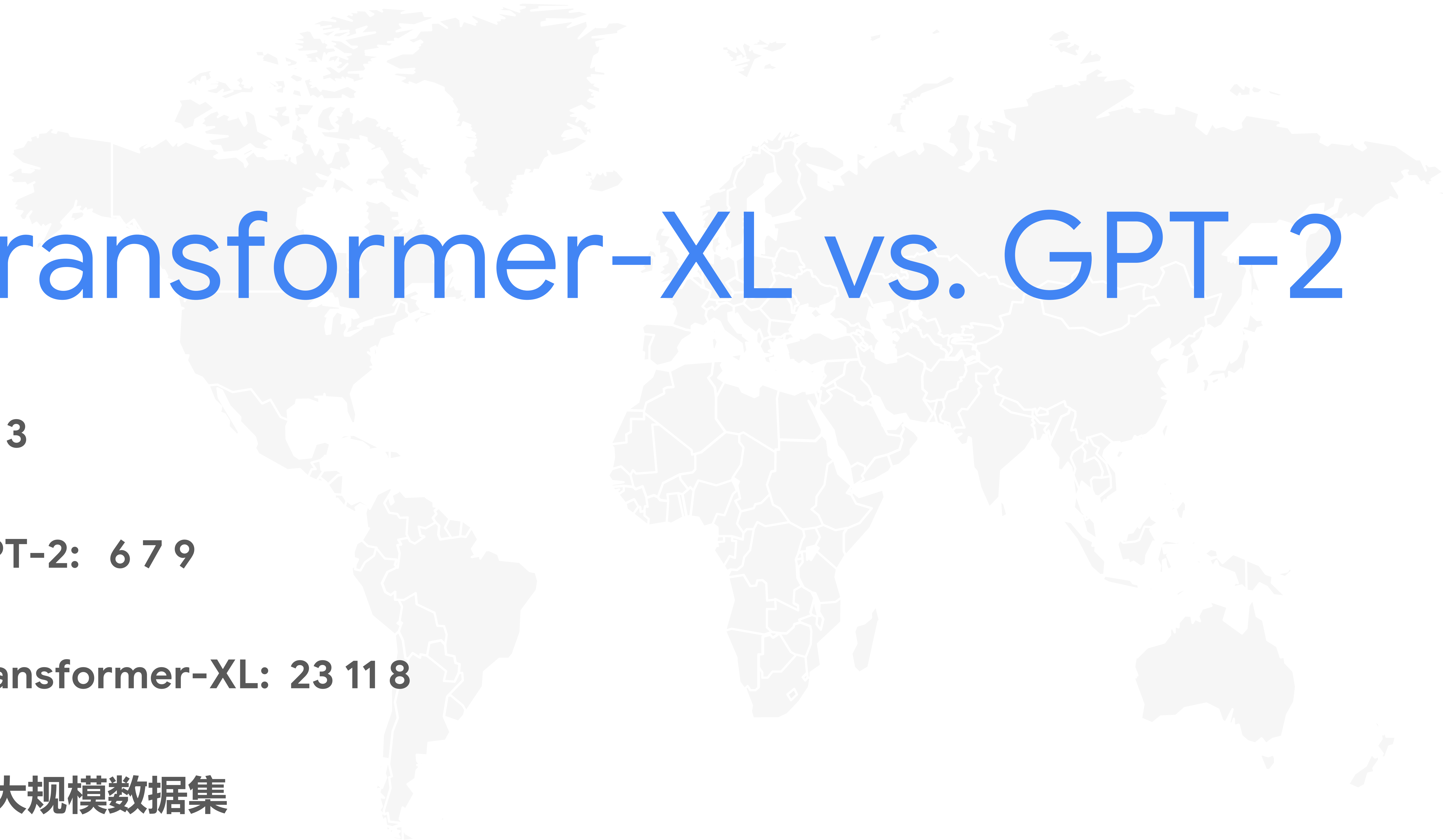
数据集

https://magenta.tensorflow.org/datasets/

MAESTRO

- MAESTRO (MIDI and Audio Edited for Synchronous TRacks and Organization) is a dataset composed of over 200 hours of virtuosic piano performances captured with fine alignment (~3 ms) between note labels and audio waveforms.

Transformer-XL vs. GPT-2

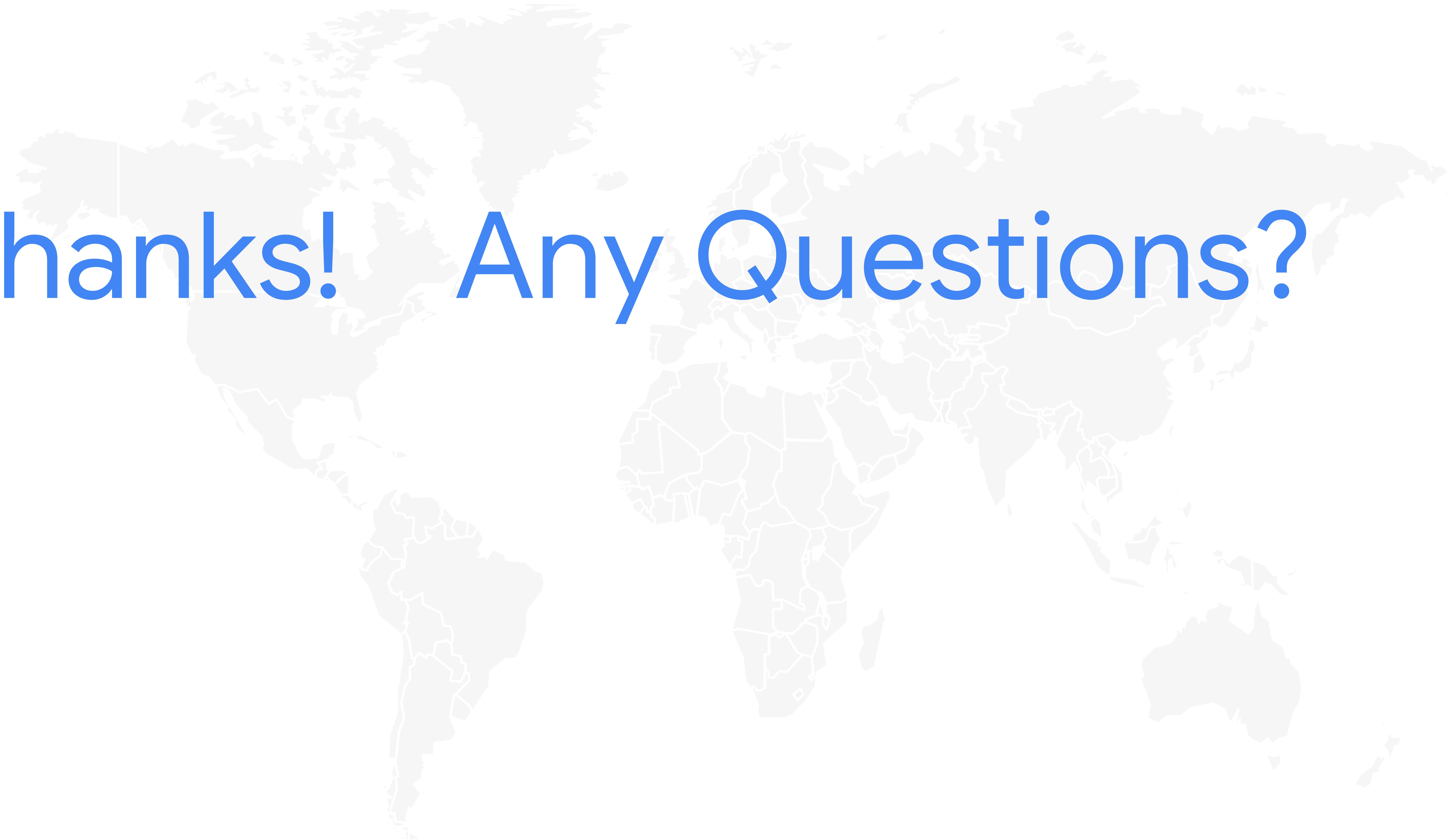


1 2 3

GPT-2: 6 7 9

Transformer-XL: 23 11 8

更大规模数据集



Thanks! Any Questions?