Advanced Deep Learning

Section 3:

Natural Language Processing

Rényi Alfréd Matematikai Kutatóintézet

ELTE EÖTVÖS LORÁND TUDOMÁNYEGYETEM

Schedule

Lectures on NLP:

- **Lecture 1: Introduction and Foundations**
  - Characteristics of the domain
  - Classical methods
  - Character encodings
  - Tokenization
  - Embeddings
- **Lecture 2: Language Models and Language Modeling**
  - Objective functions
  - Sequential modeling
  - Decoding strategies
  - Models: Transformers (BERT, GPT)
  - Training: pre-training, fine-tuning
  - Evaluation
- **Lecture 3: Large Language Models (LLMs)**
  - Emergent properties
  - Scaling laws
  - GPT-series
  - Instruction tuning
  - Reinforcement Learning with Human Feedback (RLHF)
- **Lecture 4: Research**
  - Prompt engineering
  - Multimodality: CLIP
  - Problems: hallucination
  - Retrieval Augmented Generation (RAG)
  - Security

Language Models
and
Language Modeling

Lecture 2

What is this lecture about?

This lecture tries to answer the following questions:

- What are the weaknesses of previous sequence models that needed to be solved?
- What is the basic mechanism behind the novel Transformer architecture?
- What are the advantages of Transformers compared to other approaches?
- How are LMs trained, what is the objective?
- How can we evaluate performance when for some NLP tasks the desired output is frequently ill-defined?
- What is the temperature parameter in the ChatGPT?

Language Modeling

Next Word Prediction

Language Models: Usage

Demo with LLaMa

What is Language Modeling?

Language Modeling:

- a fundamental task in NLP
- developing a model to predict the probability distribution of sequences of words in a language

Formula:

- $P(W) = P(w_1, w_2, ..., w_n) = \prod_{i=1}^{n} P(w_i | w_1, w_2, ..., w_{i-1})$
    - W: sentence / sequence
    - $w_1, w_2, ..., w_n$: words
    - $P(w_i | w_1, w_2, ..., w_{i-1})$: the probability if the i-th word given all the previous words

Objective functions

Language Modeling (LM):

- Next Word Prediction
- autoregressive

Masked Language Modeling (MLM):

- auto-encoding

Complementary Objectives:

- Next Sentence Prediction (NSP)
- Sentence Order Prediction (SOP)

Multimodal Objectives:

- Masked Visual-Language Modeling (MVLM)
- Text-Image Alignment (TIA)
- Text-Image Matching (TIM)

Objective functions

Language Modeling (LM):

- Next Word Prediction
- autoregressive

Masked Language Modeling (MLM):

- auto-encoding

Complementary Objectives:

- Next Sentence Prediction (NSP)
- Sentence Order Prediction (SOP)

Multimodal Objectives:

- Masked Visual-Language Modeling (MVLM)
- Text-Image Alignment (TIA)
- Text-Image Matching (TIM)

Objective functions

Language Modeling (LM):

- Next Word Prediction
- autoregressive

Masked Language Modeling (MLM):

- auto-encoding

Complementary Objectives:

- Next Sentence Prediction (NSP)
- Sentence Order Prediction (SOP)

Multimodal Objectives:

- Masked Visual-Language Modeling (MVLM)
- Text-Image Alignment (TIA)
- Text-Image Matching (TIM)

Examples:

- Birds fly ____
- Dogs bark at ____
- He read the morning ____
- She added sugar to her ____
- The children played outside in the ____

Objective functions

Language Modeling (LM):

- Next Word Prediction
- autoregressive

Masked Language Modeling (MLM):

- auto-encoding

Complementary Objectives:

- Next Sentence Prediction (NSP)
- Sentence Order Prediction (SOP)

Multimodal Objectives:

- Masked Visual-Language Modeling (MVLM)
- Text-Image Alignment (TIA)
- Text-Image Matching (TIM)

Objective functions

Language Modeling (LM):

- Next Word Prediction
- autoregressive

Masked Language Modeling (MLM):

- auto-encoding

Complementary Objectives:

- Next Sentence Prediction (NSP)
- Sentence Order Prediction (SOP)

Multimodal Objectives:

- Masked Visual-Language Modeling (MVLM)
- Text-Image Alignment (TIA)
- Text-Image Matching (TIM)

Examples:

- Birds fly ____
- Dogs ____ at strangers.
- He read ____ morning paper.
- She ____ sugar to her ____
- The children ____ outside in the ____

Objective functions

Language Modeling (LM):

- Next Word Prediction
- autoregressive

Masked Language Modeling (MLM):

- auto-encoding

Complementary Objectives:

- Next Sentence Prediction (NSP)
- Sentence Order Prediction (SOP)

Multimodal Objectives:

- Masked Visual-Language Modeling (MVLM)
- Text-Image Alignment (TIA)
- Text-Image Matching (TIM)

Objective functions

Language Modeling (LM):

- Next Word Prediction
- autoregressive

Masked Language Modeling (MLM):

- auto-encoding

Complementary Objectives:

- **Next Sentence Prediction (NSP)**
- Sentence Order Prediction (SOP)

Multimodal Objectives:

- Masked Visual-Language Modeling (MVLM)
- Text-Image Alignment (TIA)
- Text-Image Matching (TIM)

Objective functions

Language Modeling (LM):

- Next Word Prediction
- autoregressive

Masked Language Modeling (MLM):

- auto-encoding

Complementary Objectives:

- **Next Sentence Prediction (NSP)**:
- Sentence Order Prediction (SOP)

Multimodal Objectives:

- Masked Visual-Language Modeling (MVLM)
- Text-Image Alignment (TIA)
- Text-Image Matching (TIM)

Examples:

- A: The restaurant was incredibly busy tonight.
- B: They waited over an hour for their food.

- A: The library closed at 8 PM.
- B: She was ready for the early meeting.

Objective functions

Language Modeling (LM):

- Next Word Prediction
- autoregressive

Masked Language Modeling (MLM):

- auto-encoding

Complementary Objectives:

- Next Sentence Prediction (NSP)
- **Sentence Order Prediction (SOP)**:

Multimodal Objectives:

- Masked Visual-Language Modeling (MVLM)
- Text-Image Alignment (TIA)
- Text-Image Matching (TIM)

Objective functions

Language Modeling (LM):

- Next Word Prediction
- autoregressive

Masked Language Modeling (MLM):

- auto-encoding

Complementary Objectives:

- Next Sentence Prediction (NSP)
- **Sentence Order Prediction (SOP)**:

Multimodal Objectives:

- Masked Visual-Language Modeling (MVLM)
- Text-Image Alignment (TIA)
- Text-Image Matching (TIM)

Examples:

- A: She posted it online.
- B: She took a photo.

- A: Why don’t scientists trusts atoms?
- B: Because they make up everything.

Objective functions

Language Modeling (LM):

- Next Word Prediction
- autoregressive

Masked Language Modeling (MLM):

- auto-encoding

Complementary Objectives:

- Next Sentence Prediction (NSP)
- Sentence Order Prediction (SOP)

Multimodal Objectives:

- Masked Visual-Language Modeling (MVLM)
- Text-Image Alignment (TIA)
- Text-Image Matching (TIM)

Objective functions

Language Modeling (LM):

- Next Word Prediction
- autoregressive

Masked Language Modeling (MLM):

- auto-encoding

Complementary Objectives:

- Next Sentence Prediction (NSP)
- Sentence Order Prediction (SOP)

Multimodal Objectives:

- **Masked Visual-Language Modeling (MVLM)**:
- **Text-Image Alignment (TIA)**:
- **Text-Image Matching (TIM)**:

Objective functions

Language Modeling (LM):

- Next Word Prediction
- autoregressive

Masked Language Modeling (MLM):

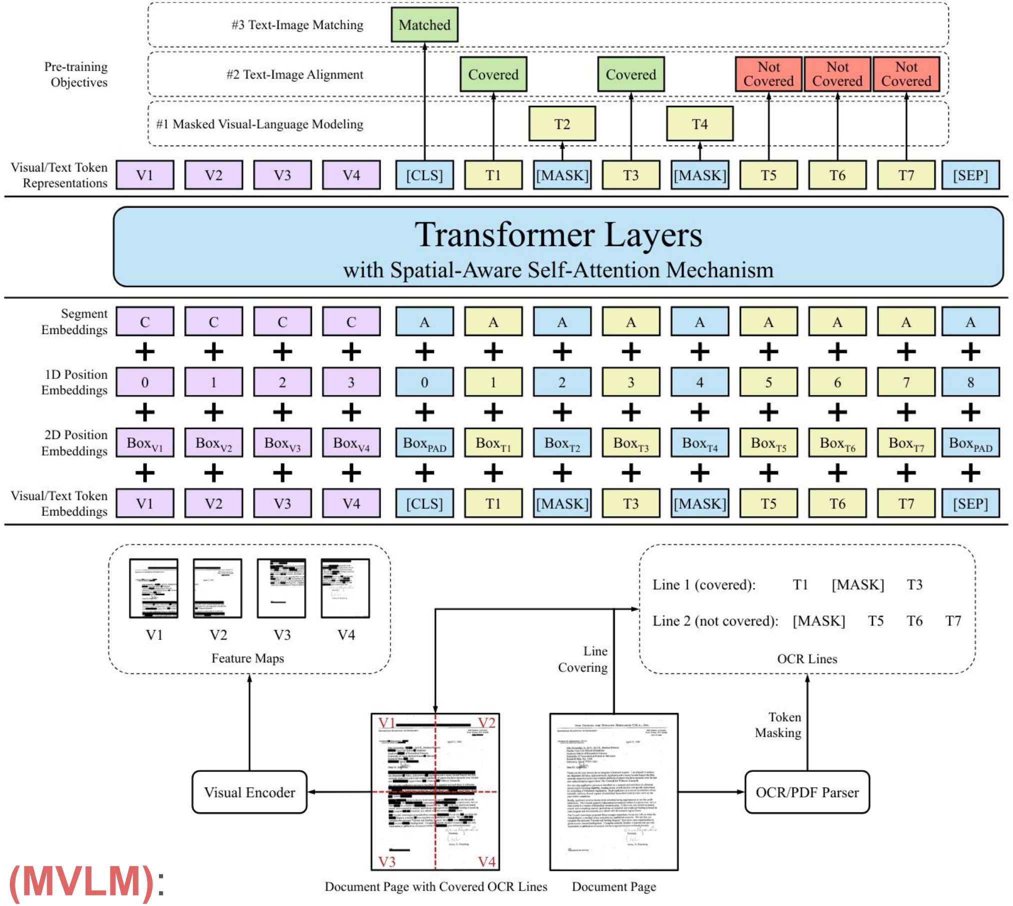
- auto-encoding

Complementary Objectives:

- Next Sentence Prediction (NSP)
- Sentence Order Prediction (SOP)

Multimodal Objectives:

- **Masked Visual-Language Modeling (MVLM)**:
- **Text-Image Alignment (TIA)**:
- **Text-Image Matching (TIM)**:



Pre-training Objectives
#3 Text-Image Matching
Matched
#2 Text-Image Alignment
Covered
Covered
Not Covered
Not Covered
Not Covered
#1 Masked Visual-Language Modeling
T2
T4
Visual/Text Token Representations
V1 V2 V3 V4 [CLS] T1 [MASK] T3 [MASK] T5 T6 T7 [SEP]
Transformer Layers
with Spatial-Aware Self-Attention Mechanism
Segment Embeddings
C C C C A A A A A A A A A
1D Position Embeddings
0 1 2 3 0 1 2 3 4 5 6 7 8
2D Position Embeddings
Box_{V1} Box_{V2} Box_{V3} Box_{V4} Box_{PAD} Box_{T1} Box_{T2} Box_{T3} Box_{T4} Box_{T5} Box_{T6} Box_{T7} Box_{PAD}
Visual/Text Token Embeddings
V1 V2 V3 V4 [CLS] T1 [MASK] T3 [MASK] T5 T6 T7 [SEP]
V1 V2 V3 V4
Feature Maps
Line 1 (covered): T1 [MASK] T3
Line 2 (not covered): [MASK] T5 T6 T7
OCR Lines
Line Covering
Token Masking
Visual Encoder
OCR/PDF Parser
V1 V2 V3 V4
Document Page with Covered OCR Lines
Document Page

Objective functions

Language Modeling (LM):

- Next Word Prediction
- autoregressive

Masked Language Modeling (MLM):

- auto-encoding

Complementary Objectives:

- Next Sentence Prediction (NSP)
- Sentence Order Prediction (SOP)

Multimodal Objectives:

- Masked Visual Language Modeling (MVLM)
- Text-Image Alignment (TIA)
- Text-Image Matching (TIM)

Why the Language Modeling (LM) objective is so good?

Language Modeling (LM): What we learn when predicting next word?

- Goal: predicting the next word (token)

Examples:

- 1. Pattern / Frequency:
    - frequently used word connections, phrases
    - Example: “One two three ____.” (four)
    - Notions:
        - n-gram
        - language modeling
- 2. Memorization:
    - a well known text part
    - there can be a lot → large memory
    - Example: “To be or not to be, that is the _____.” (question)
    - Notions:
        - memory, free-parameters, network size
- 3. Understand meaning, concepts:
    - relations between words
    - identifying parts / words which are important
    - “He was known for his punctuality, yet today he arrived at the meeting ___.” (late)
    - Notions:
- 4. Interpreting intention:
    - catching intent
    - “Here is a sentence where every second word is ‘egg’: Where ____” (egg)

Decoding Strategies

Model output:

- $P(w_i|w_1, w_2, ..., w_{i-1})$:
  - a probability distribution over the all the words
  - there is no one solution - there can be more valid continuations

Conditional probability distribution

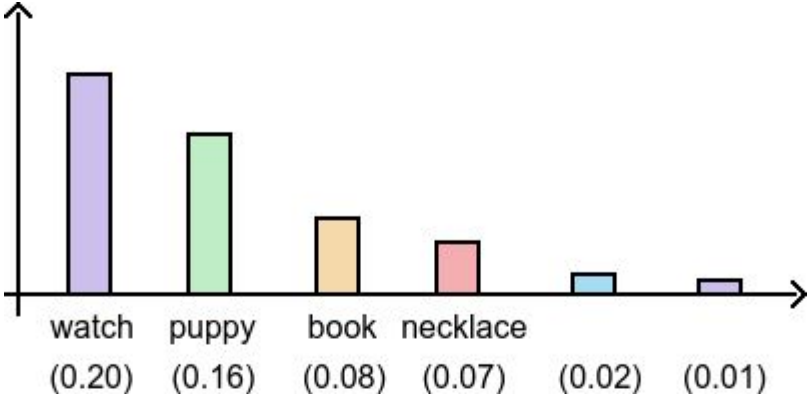
Example:

- “She opened her birthday gift and found a ____”
- P(w_i | “She”, “opened”, “her”, “birthday”, “gift”, “and”, “found”, “a”)

Decoding Strategies

Model output:

- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**



She opened her birthday gift and found a …

Decoding Strategies

Considerations:

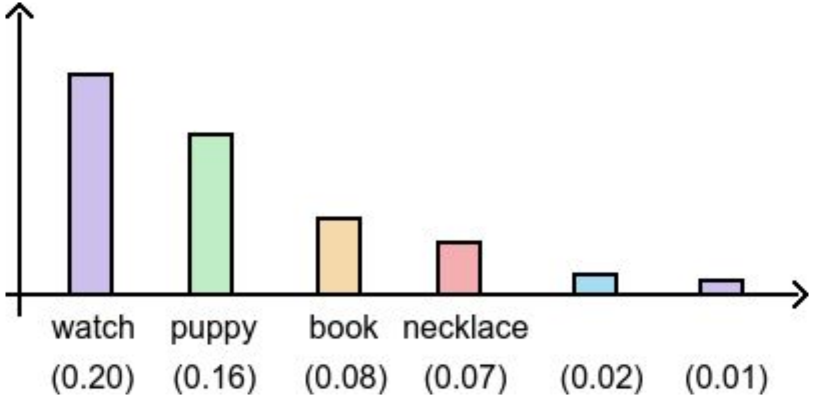
- plausible output
- diversity
- not too long
- different outputs for repeated inference

She opened her birthday gift and found a …

Decoding Strategies

Model output:

- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**

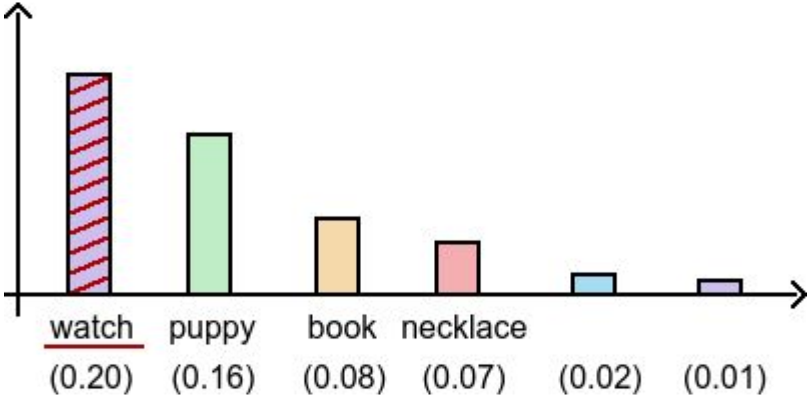


She opened her birthday gift and found a …

Decoding Strategies

Model output:

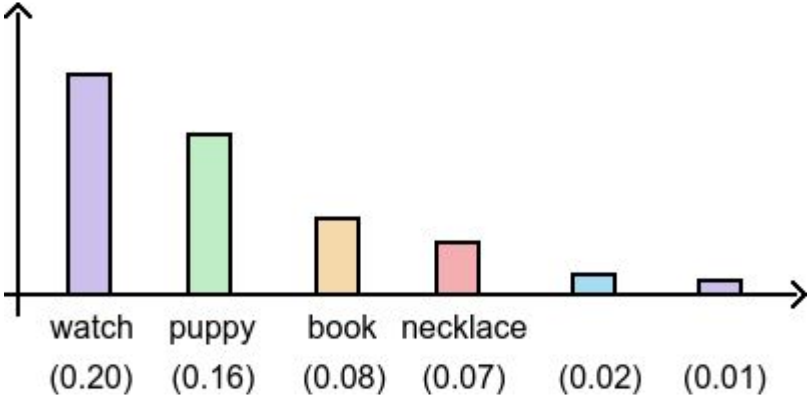
- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**



Decoding Strategies

Model output:

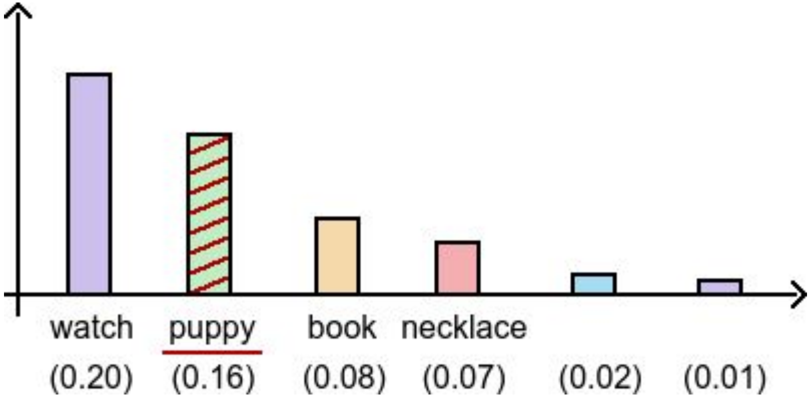
- How to pick?
  - Deterministic
  - Random sampling
  - Temperature-scaled random sampling
  - k-sampling
  - p-sampling
  - Beam search



Decoding Strategies

Model output:

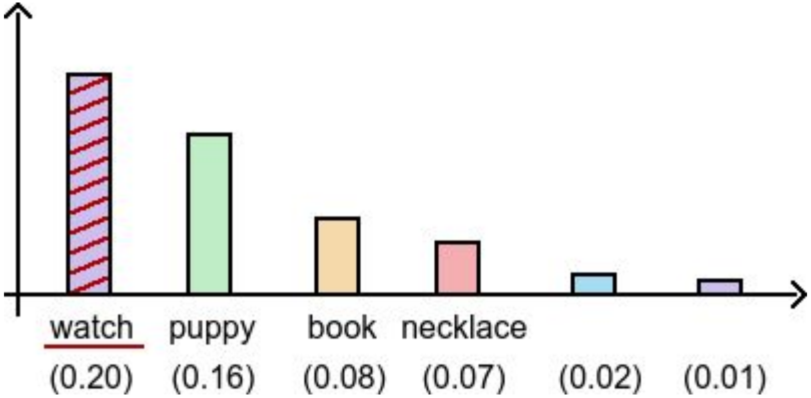
- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**



Decoding Strategies

Model output:

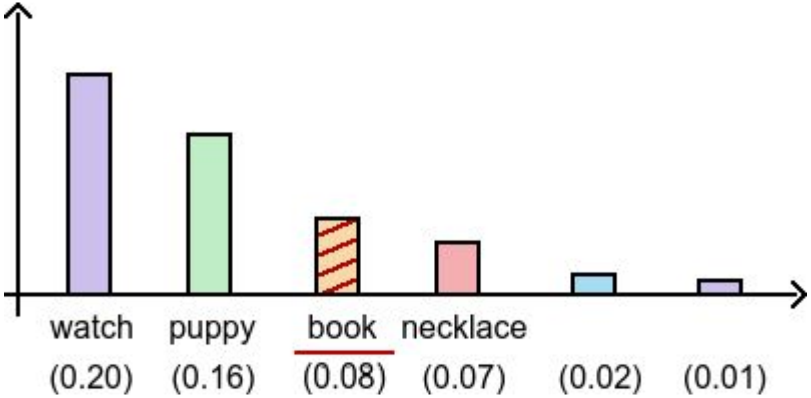
- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**



Decoding Strategies

Model output:

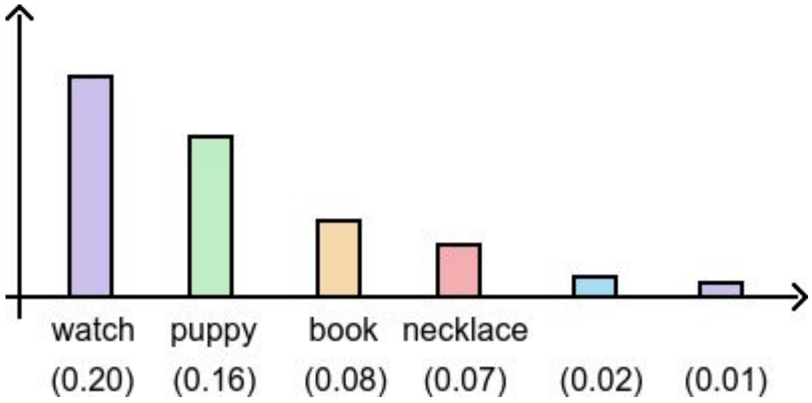
- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**



Decoding Strategies

Model output:

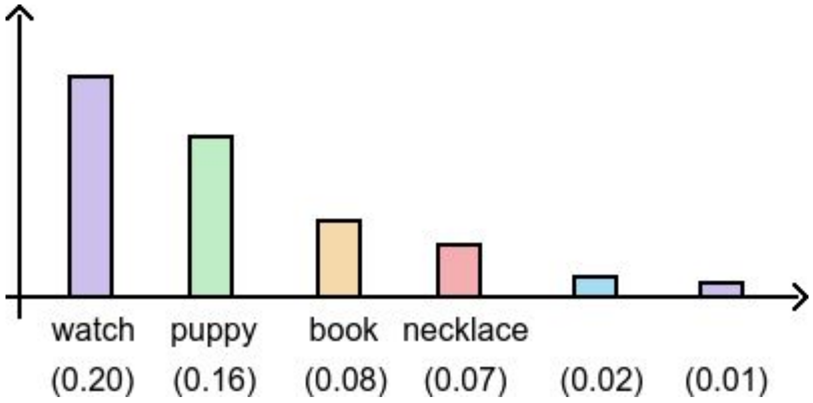
- How to pick?
    - **Deterministic**
    - **Random sampling**
    - **Temperature-scaled random sampling**
    - **k-sampling**
    - **p-sampling**
    - **Beam search**

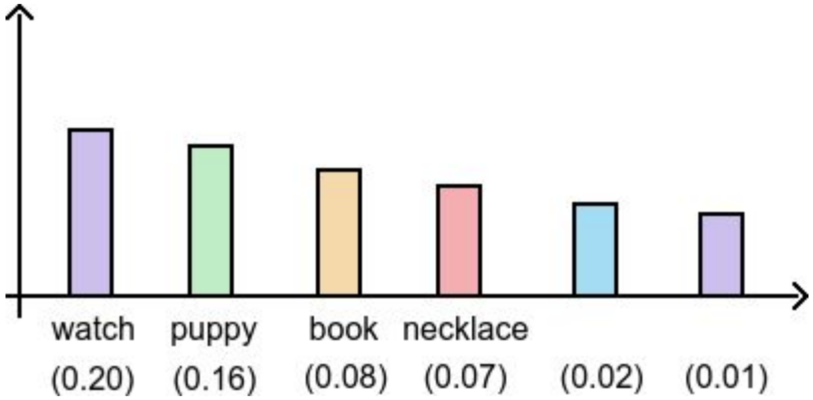


Decoding Strategies

Model output:

- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**

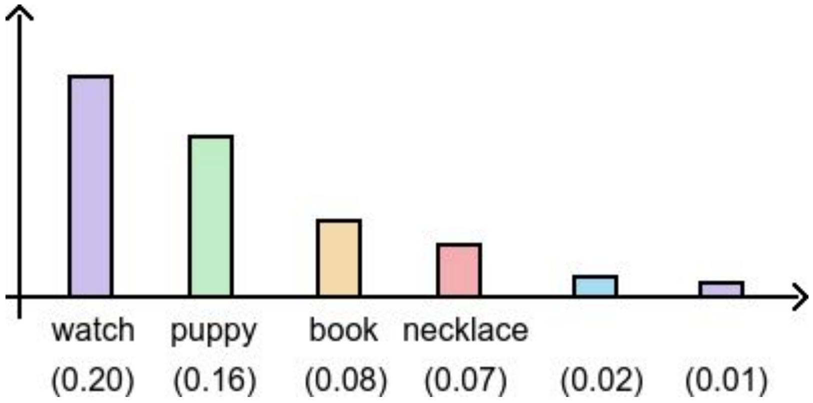


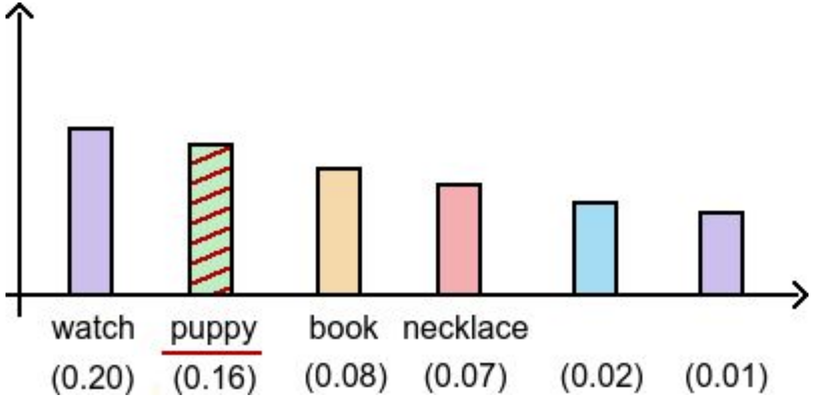


Decoding Strategies

Model output:

- How to pick?
  - Deterministic
  - Random sampling
  - Temperature-scaled random sampling
  - k-sampling
  - p-sampling
  - Beam search

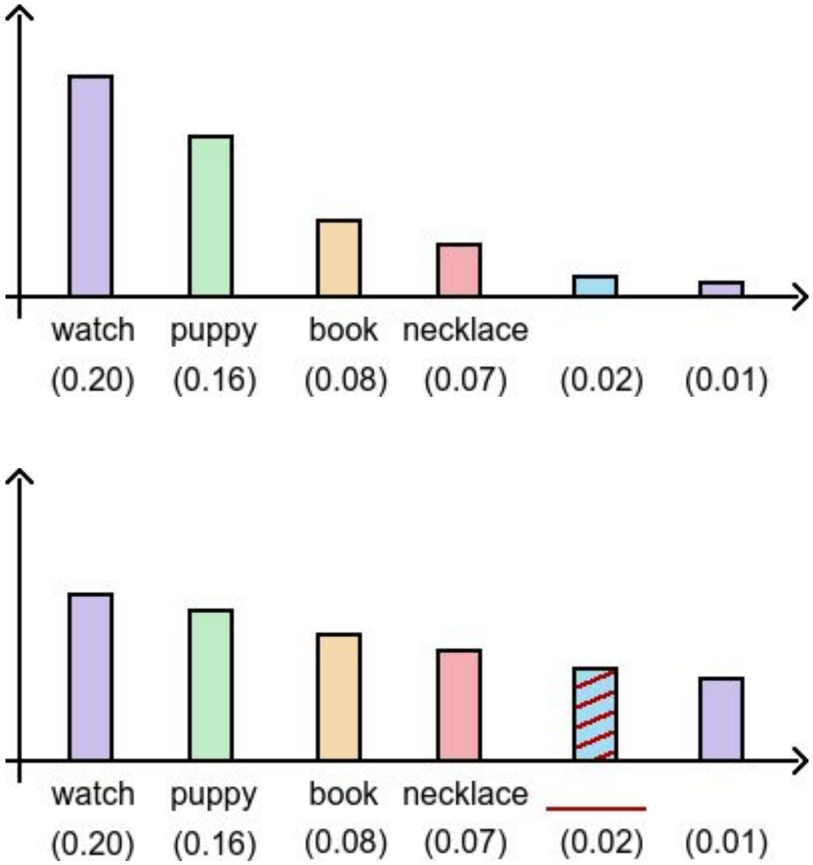




Decoding Strategies

Model output:

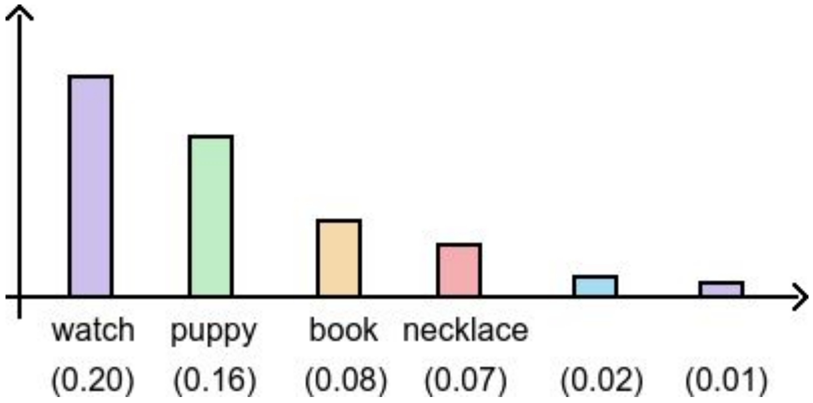
- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**



Decoding Strategies

Model output:

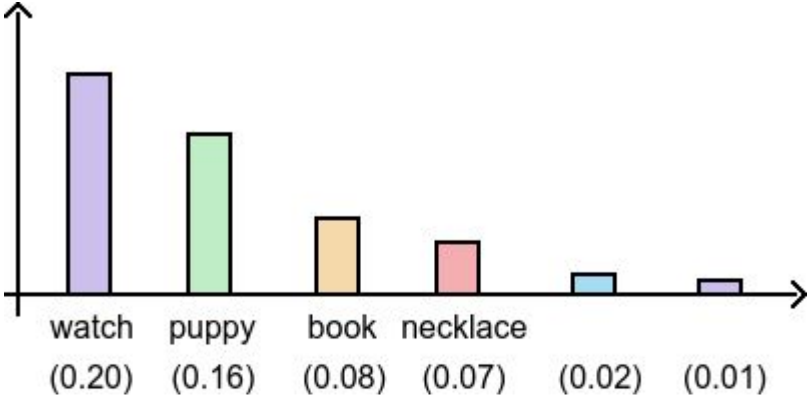
- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**

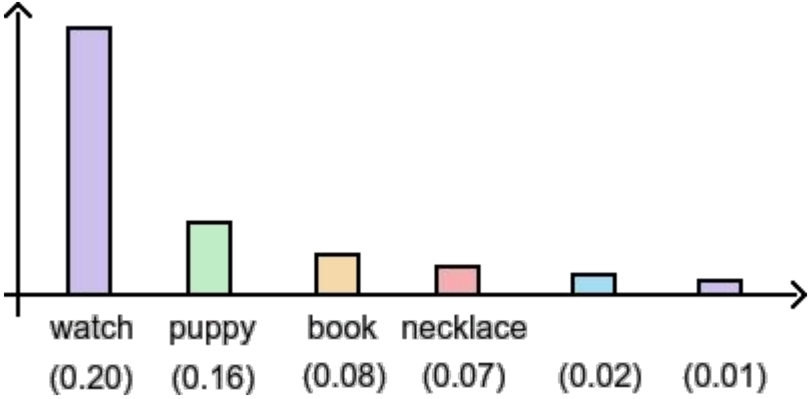


Decoding Strategies

Model output:

- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**

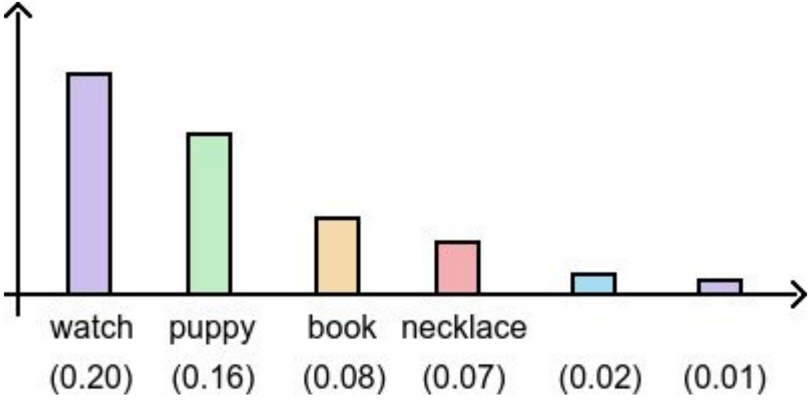


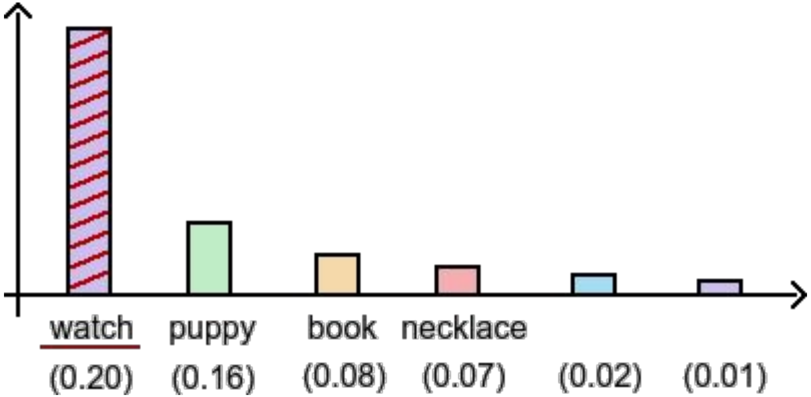


Decoding Strategies

Model output:

- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**

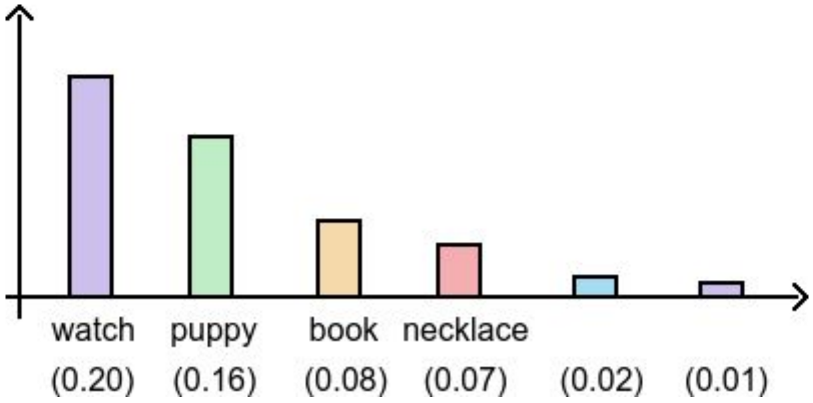




Decoding Strategies

Model output:

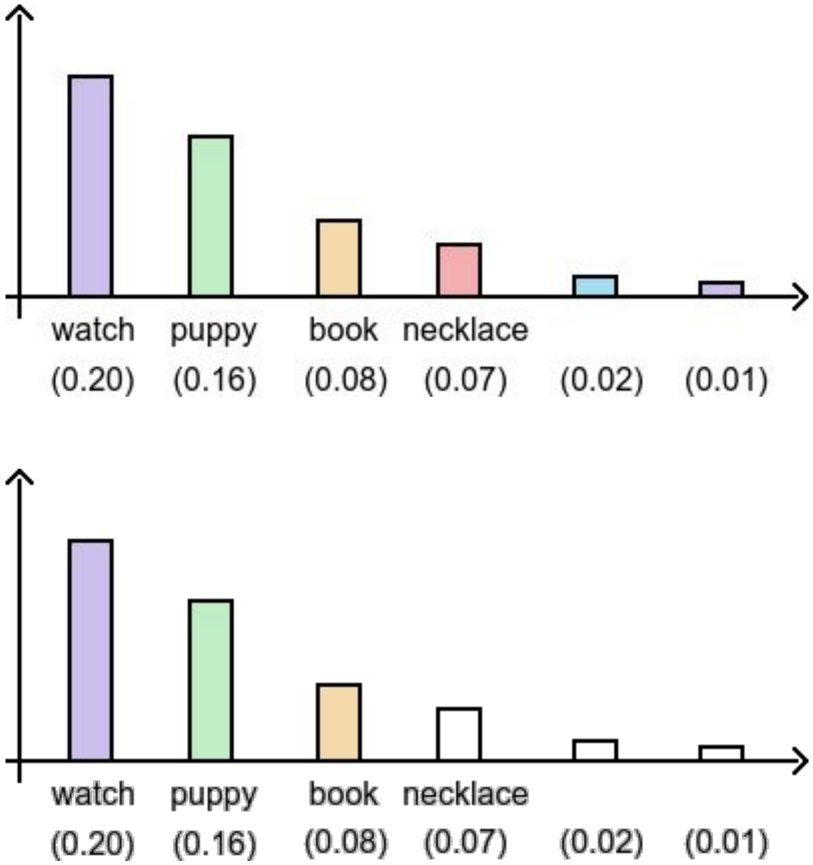
- How to pick?
    - **Deterministic**
    - **Random sampling**
    - **Temperature-scaled random sampling**
    - **k-sampling**
    - **p-sampling**
    - **Beam search**



Decoding Strategies

Model output:

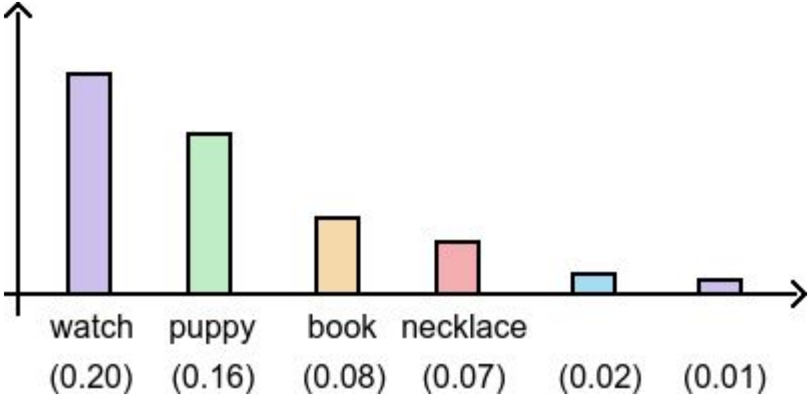
- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**



Decoding Strategies

Model output:

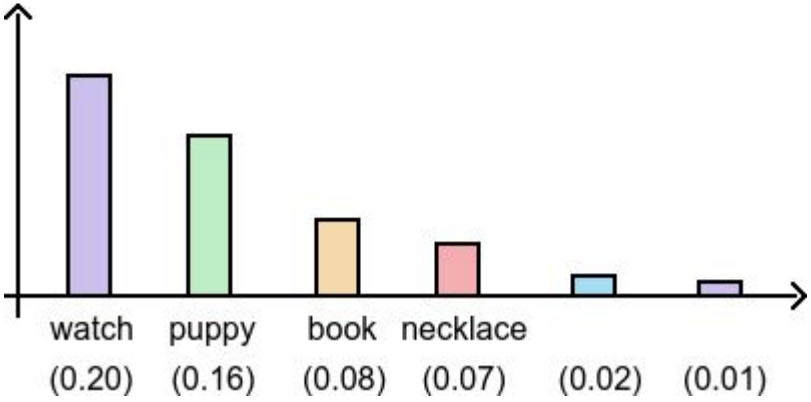
- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**

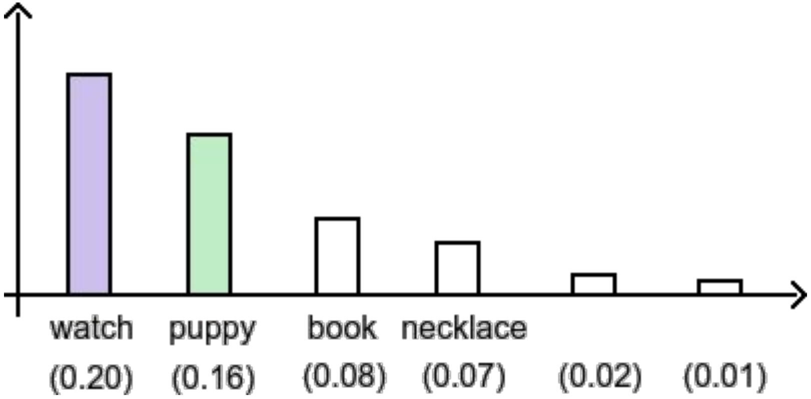


Decoding Strategies

Model output:

- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**

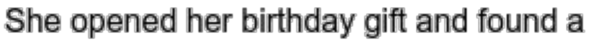




Decoding Strategies

Model output:

- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**

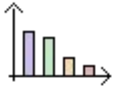


She opened her birthday gift and found a

Decoding Strategies

Model output:

- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**

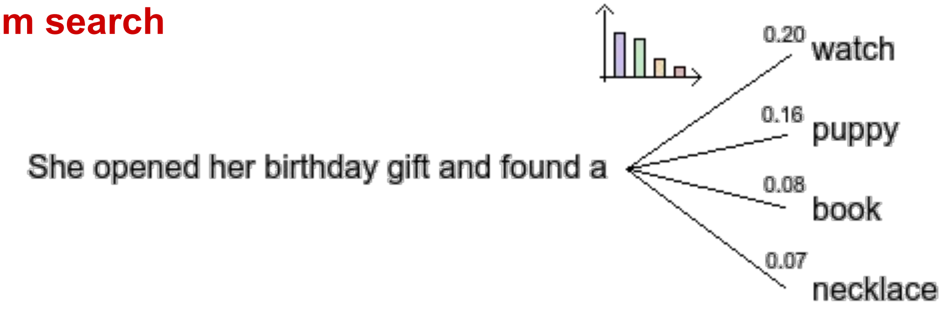


She opened her birthday gift and found a

Decoding Strategies

Model output:

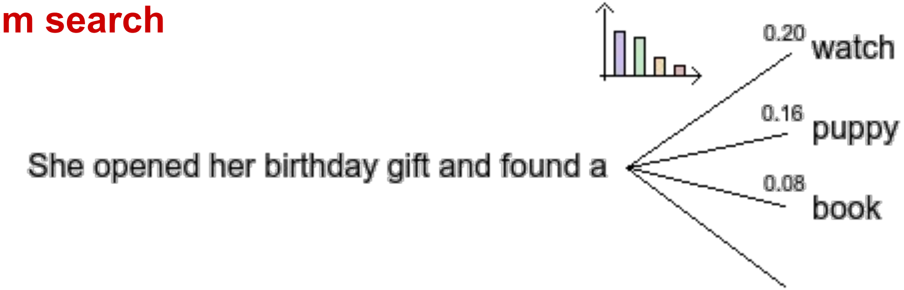
- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**



Decoding Strategies

Model output:

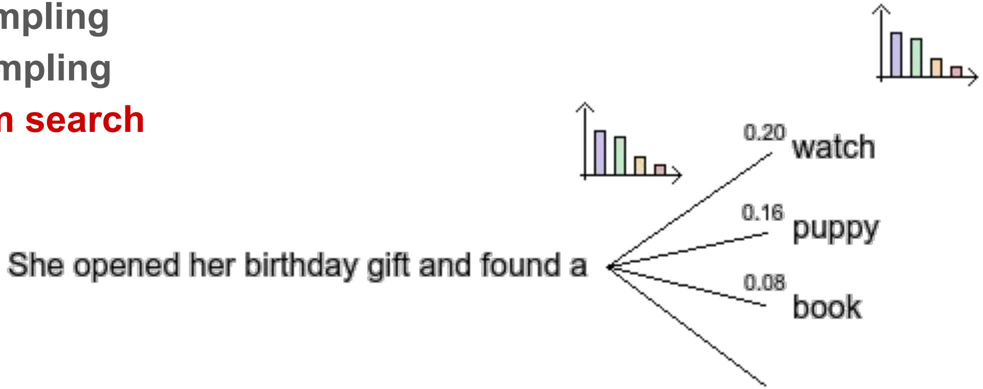
- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**



Decoding Strategies

Model output:

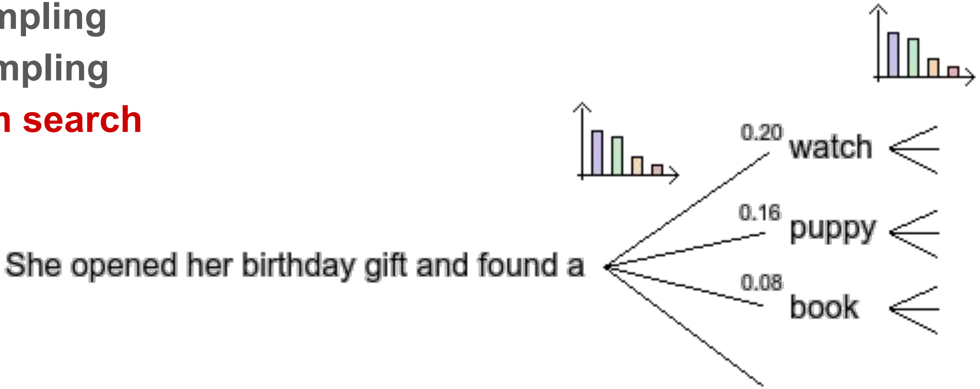
- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**



Decoding Strategies

Model output:

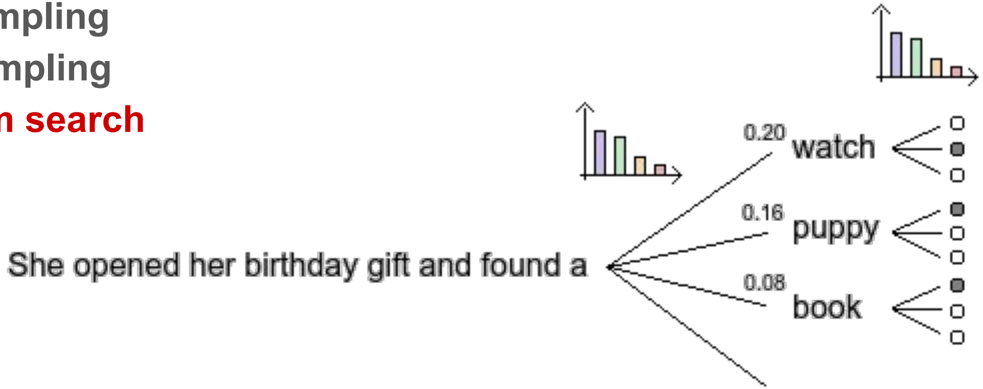
- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**



Decoding Strategies

Model output:

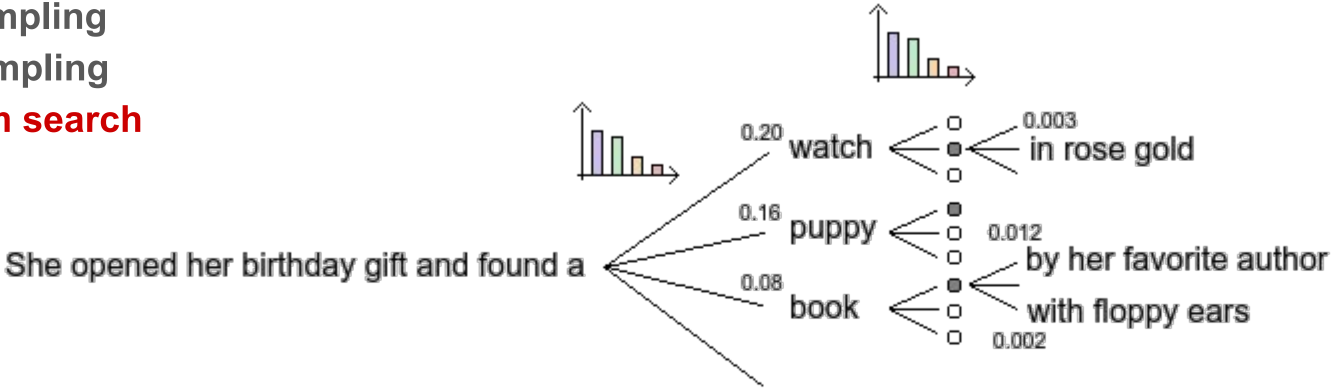
- How to pick?
    - **Deterministic**
    - **Random sampling**
    - **Temperature-scaled random sampling**
    - **k-sampling**
    - **p-sampling**
    - **Beam search**



Decoding Strategies

Model output:

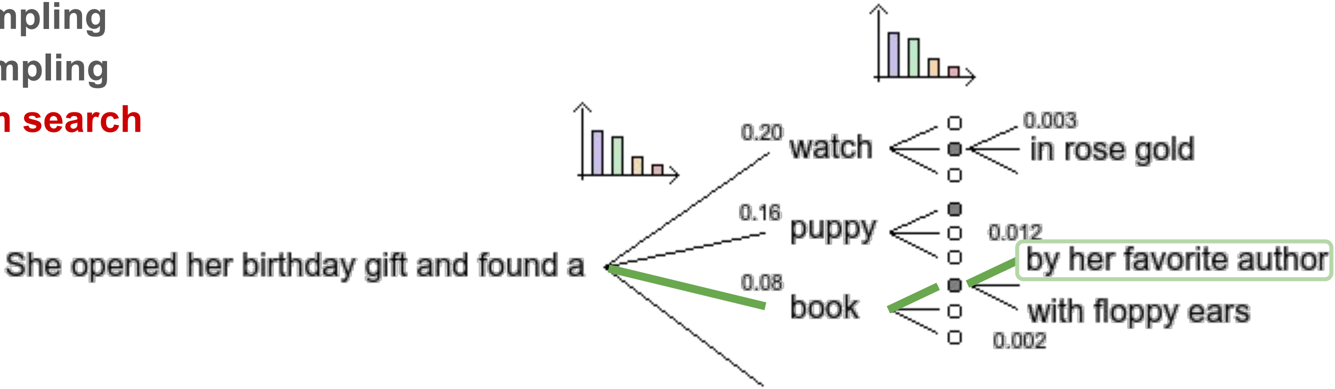
- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**



Decoding Strategies

Model output:

- How to pick?
  - **Deterministic**
  - **Random sampling**
  - **Temperature-scaled random sampling**
  - **k-sampling**
  - **p-sampling**
  - **Beam search**



Task we can solve by Language Modeling

Task we can solve by Language Modeling

Text Generation

Question Answering

Machine Translation

Summarization

Chatbots

Unsupervised / Supervised / Self-Supervised Learning

Unsupervised Learning:

- there is a lot of data
- Examples:
    - clustering, feature extraction, compression

Supervised Learning:

- rich training signal
- expensive to create
- Examples:
    - classification, regression

Self-Supervised Learning:

- creating labels (supervision) from unsupervised data
- best of both

Unsupervised / Supervised / Self-Supervised Learning

Unsupervised Learning:

- there is a lot of data
- Examples:
    - clustering, feature extraction, compression

Supervised Learning:

- rich training signal
- expensive to create
- Examples:
    - classification, regression

Self-Supervised Learning:

- creating labels (supervision) from unsupervised data
- best of both

Unsupervised / Supervised / Self-Supervised Learning

Unsupervised Learning:

- there is a lot of data
- Examples:
  - clustering, feature extraction, compression

Supervised Learning:

- rich training signal
- expensive to create
- Examples:
  - classification, regression

Self-Supervised Learning:

- creating labels (supervision) from unsupervised data
- best of both

Unsupervised / Supervised / Self-Supervised Learning

Unsupervised Learning:

- there is a lot of data
- Examples:
    - clustering, feature extraction, compression

Supervised Learning:

- rich training signal
- expensive to create
- Examples:
    - classification, regression

Self-Supervised Learning:

- creating labels (supervision) from unsupervised data
- best of both

Models

General architectures

Sequence-to-Sequence models:

- aka. seq2seq
- commonly used in Natural Language Processing
- converting sequences from one domain to sequences in another domain
- Tasks:
    - Machine Translation
    - Text Summarization
    - Question Answering
- Considerations:
    - input and output sequences have variable length
    - there can be more correct outputs
    - how do we connect the input and output sequences
        - usually a vector compresses the encoded parts

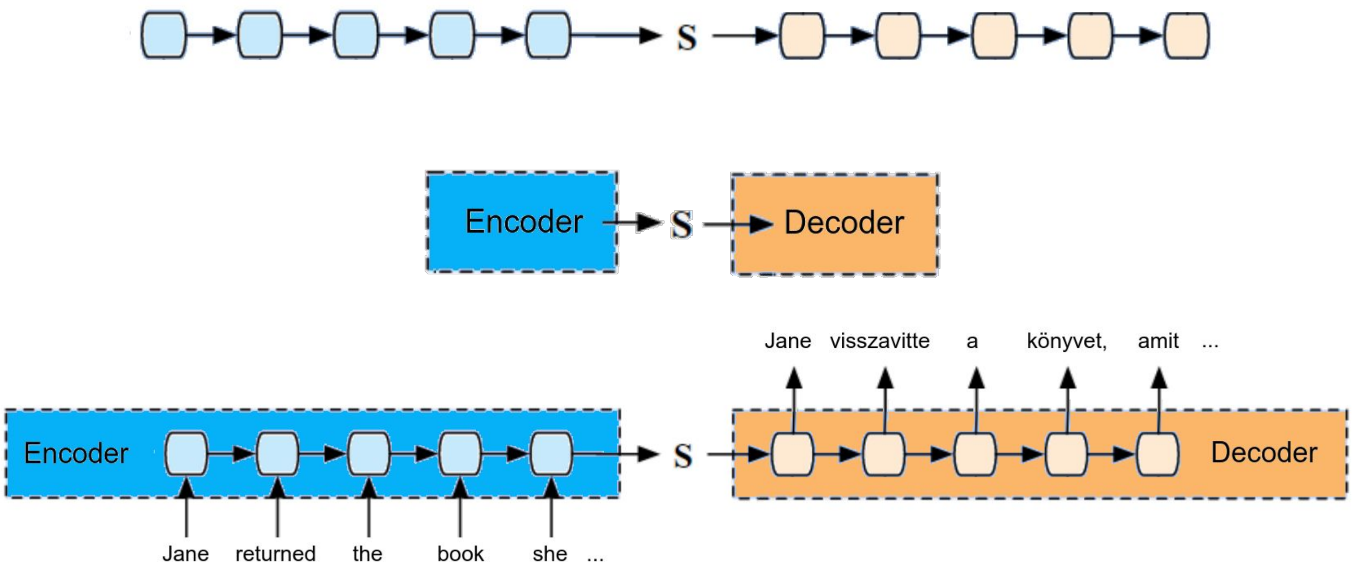
General architectures

Sequence-to-Sequence models:

- “Jane returned the book she borrowed from the library last month.”
- Jane visszavitte a könyvet, amit múlt hónapban kölcsönzött ki a könyvtárból.
- Jane visszavitte a könyvtárba a könyvet, amit a múlt hónapban kölcsönzött ki.
- A múlt hónapban kikölcsönzött könyvet Jane visszaadta a könyvtárnak.

General architectures

Sequence-to-Sequence and Encoder Decoder models:



General architectures

Sequence-to-Sequence models:

- aka. seq2seq
- commonly used in Natural Language Processing
- converting sequences from one domain to sequences in another domain
- Tasks:
    - Machine Translation
    - Summarization

Encoder-Decoder architectures:

- Encoder: extracting features
- Decoder: generating output

General architectures

Sequence-to-Sequence models:

- aka. seq2seq
- commonly used in Natural Language Processing
- converting sequences from one domain to sequences in another domain
- Tasks:
  - Machine Translation
  - Summarization

Encoder-Decoder architectures:

- Encoder: extracting features
- Decoder: generating output

General architectures

Sequence-to-Sequence models:

- aka. seq2seq
- commonly used in Natural Language Processing
- converting sequences from one domain to sequences in another domain
- Tasks:
  - Machine Translation
  - Summarization

Encoder-Decoder architectures:

- Encoder: extracting features
- Decoder: generating output

Architectures for processing sequences

Recurrent Neural Networks (RNNs):

- Processing Sequential Data
    - item-by-item
- Considerations:
    - State: representing the network’s memory
        - capturing information of the past

Architectures for processing sequences

Recurrent Neural Networks (RNNs):

- **“Plain” RNNs**
- **LSTM: Long Short-Term Memory**
- **GRU: Gated Recurrent Unit**

Architectures for processing sequences

Recurrent Neural Networks (RNNs):

- **“Plain” RNNs**:
    - designed to handle sequence data
    - maintaining a form of memory
    - a superset of various types of Recurrent Networks (e.g., LSTM, GRU)
    - States:
        - hidden state: used to retain information from the previous timesteps and to make predictions at the current timestep
    - Problems:
        - difficult to train for long sequences
            - Vanishing gradients
            - Exploding gradients
- **LSTM: Long Short-Term Memory**
- **GRU: Gated Recurrent Unit**

Architectures for processing sequences

Recurrent Neural Networks (RNNs):

- **“Plain” RNNs**
- **LSTM: Long Short-Term Memory**
    - advanced type of RNN
    - designed to avoid the long-term dependency problem
        - maintaining a longer memory
    - has a complex architecture with a system of gates that control the flow of information
        - gates can learn which data in a sequence is important to keep or disregard
        - input, output, and forget gates
    - States:
        - Hidden state: the output of the LSTM unit used for predictions and and transferred to the next timestep
        - Cell state: the internal memory of the LSTM, designed to maintain information over longer periods
            - Gates:
                - Input gate: decides what new information to add to the cell state
                - Forget gate: decides what information to discard from the cell state
                - Output gate: decides what information from the cell state to output to the hidden state
- **GRU: Gated Recurrent Unit**

Architectures for processing sequences

Recurrent Neural Networks (RNNs):

- **“Plain” RNNs**
- **LSTM: Long Short-Term Memory**
- **GRU: Gated Recurrent Unit**
    - a variation of LSTM with a simpler structure
    - comparable performance to LSTMs but computationally more efficient
    - States:
        - Hidden state:
            - Cell and Hidden states are merged
            - Gates:
                - Update gate: determines how much inf the past information needs to be passed along to the future
                - Reset gate: decides how much of the past information to forget

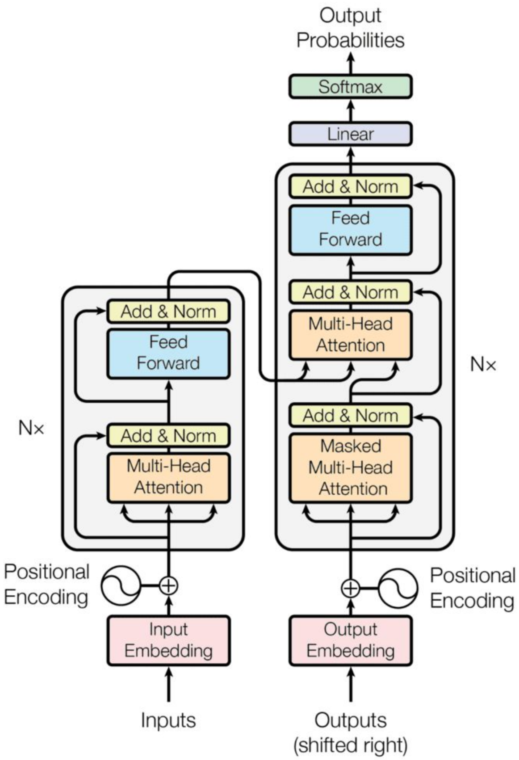
Problems of RNNs

Problems of RNNs:

- Long-term dependencies:
  - vanishing and exploding gradients
- Sequential Processing:
  - lack of parallelization
  - sequentially processes data
- O(n) to connect items in the sequence:
  - n steps to connect 2 items in the sequence having distance n between them
- Unidirectional processing:
  - bidirectional RNNs
  - shallow
- Difficulty in Capturing Contextual Information

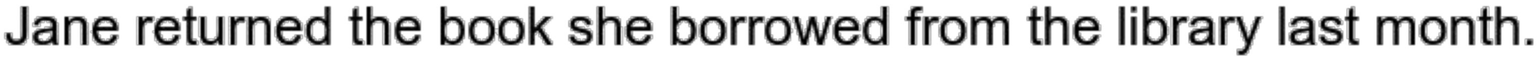
Transformer

Transformer



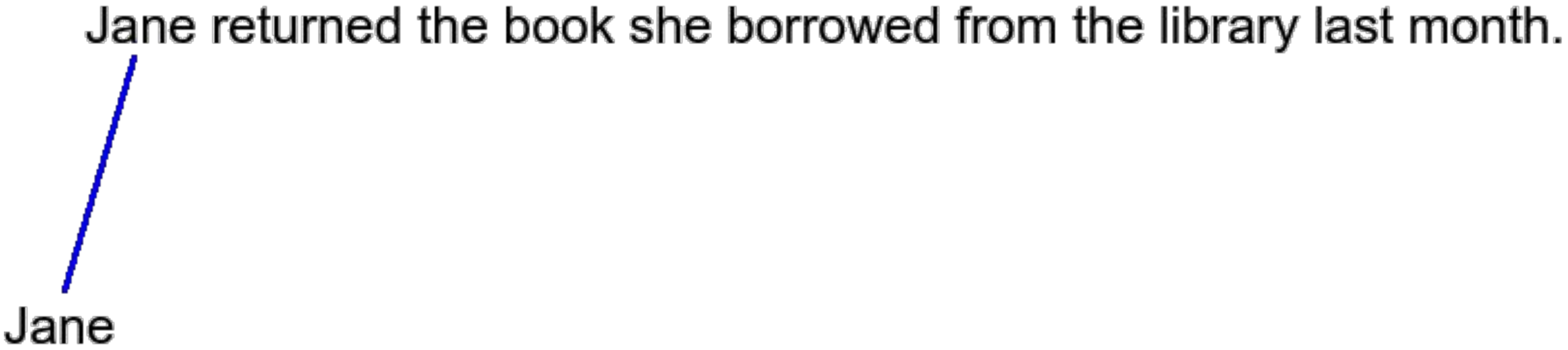
Transformer: Attention & Translation

Transformer: Attention & Translation



Jane returned the book she borrowed from the library last month.

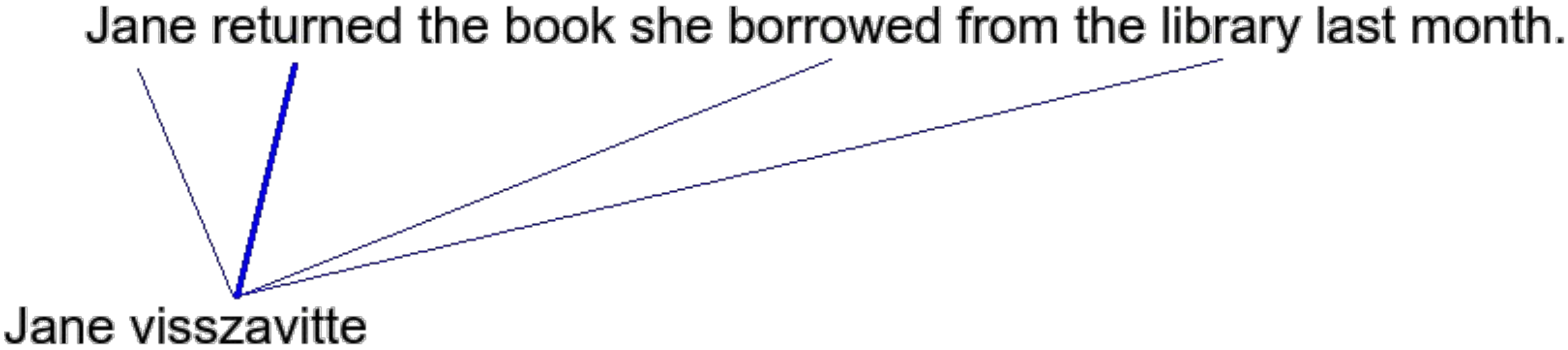
Transformer: Attention & Translation



Jane returned the book she borrowed from the library last month.

Jane

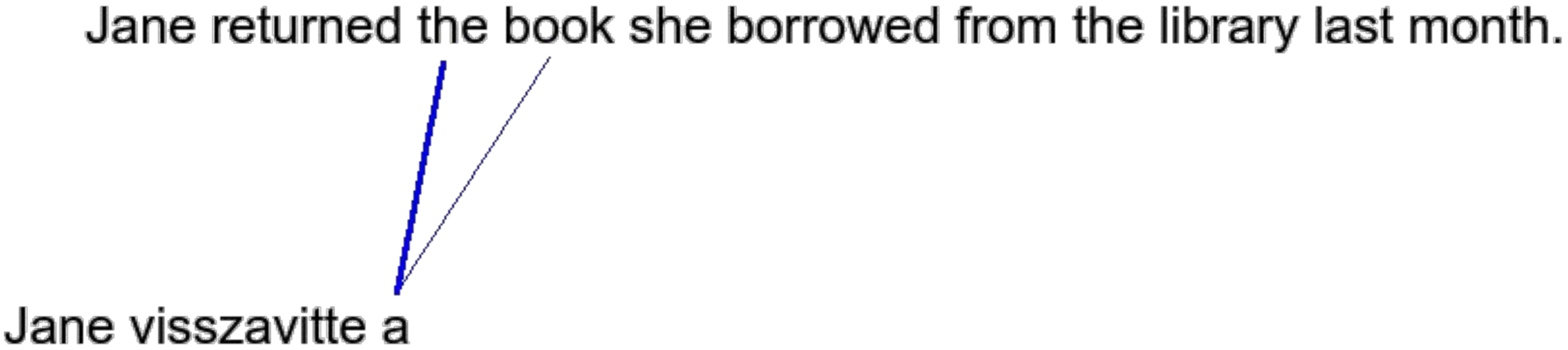
Transformer: Attention & Translation



Jane returned the book she borrowed from the library last month.

Jane visszavitte

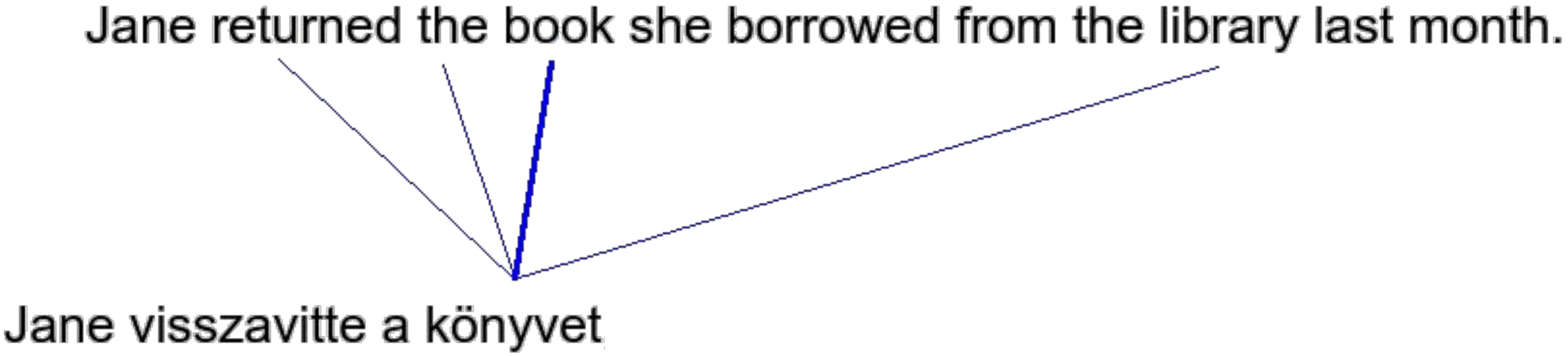
Transformer: Attention & Translation



Jane returned the book she borrowed from the library last month.

Jane visszavitte a

Transformer: Attention & Translation



Jane returned the book she borrowed from the library last month.

Jane visszavitte a könyvet

Transformer: Attention & Translation

Jane returned the book she borrowed from the library last month.

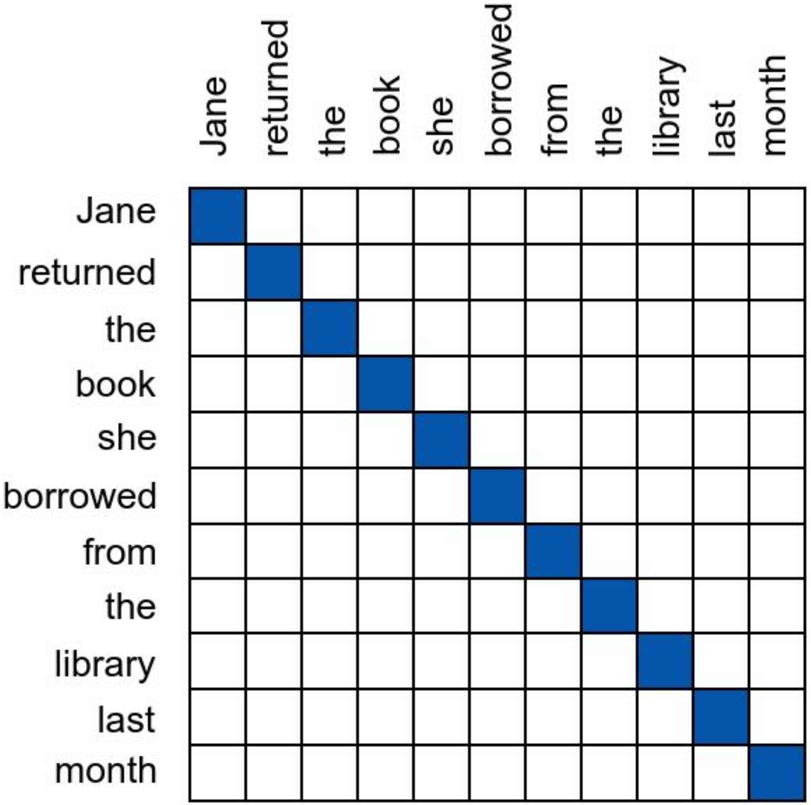
Jane visszavitte a könyvet, amit múlt hónapban kölcsönzött ki a könyvtárból.

Transformer: Attention, Self-Attention, Cross-Attention

Transformer: Attention, Self-Attention, Cross-Attention

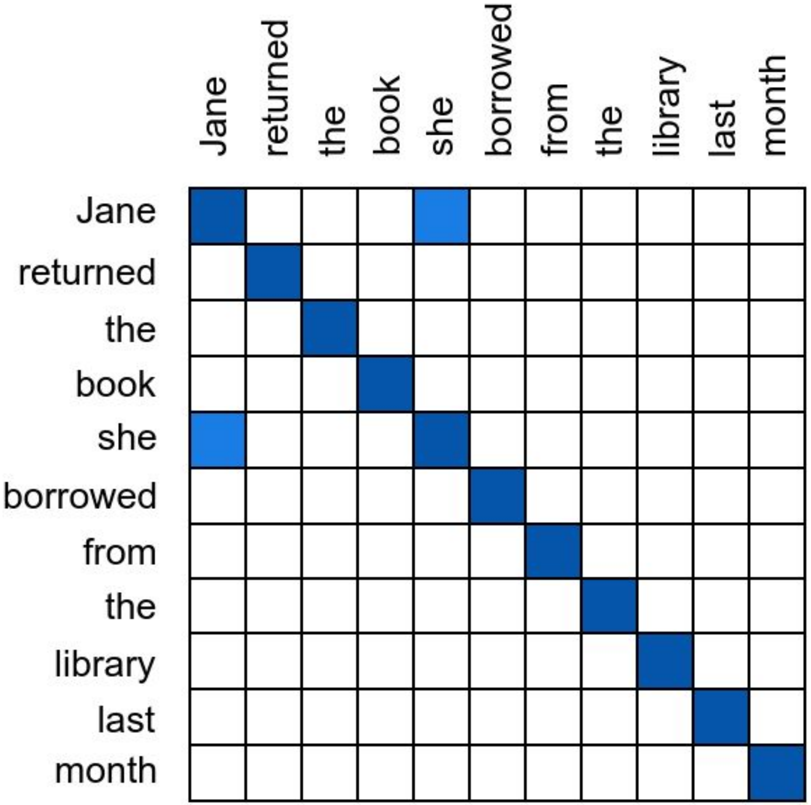
	Jane	returned	the	book	she	borrowed	from	the	library	last	month
Jane											
returned											
the											
book											
she											
borrowed											
from											
the											
library											
last											
month											

Transformer: Attention, Self-Attention, Cross-Attention

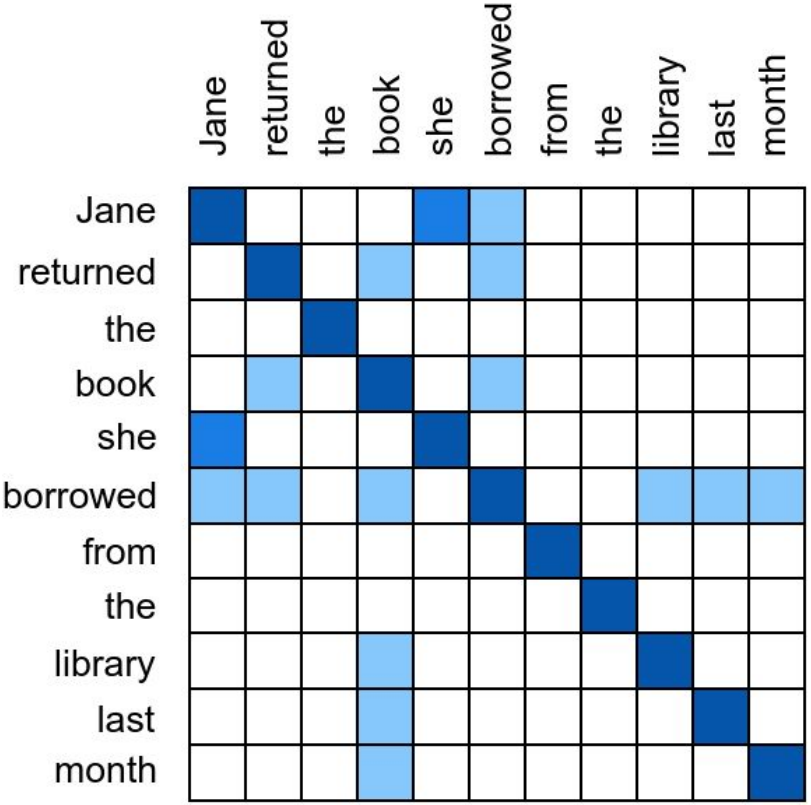


	Jane	returned	the	book	she	borrowed	from	the	library	last	month
Jane	■										
returned		■									
the			■								
book				■							
she					■						
borrowed						■					
from							■				
the								■			
library									■		
last										■	
month											■

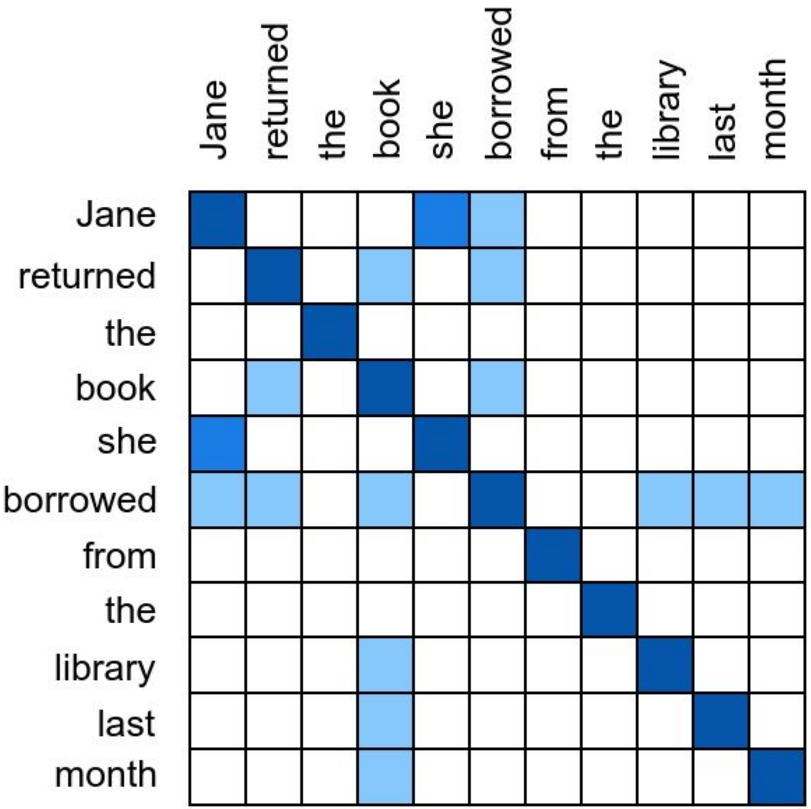
Transformer: Attention, Self-Attention, Cross-Attention

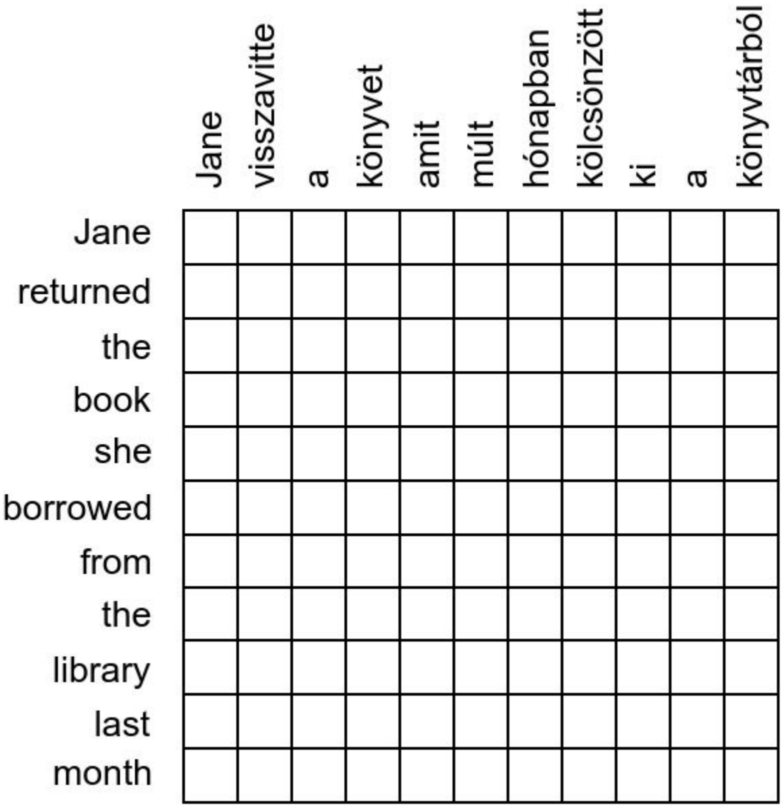


Transformer: Attention, Self-Attention, Cross-Attention

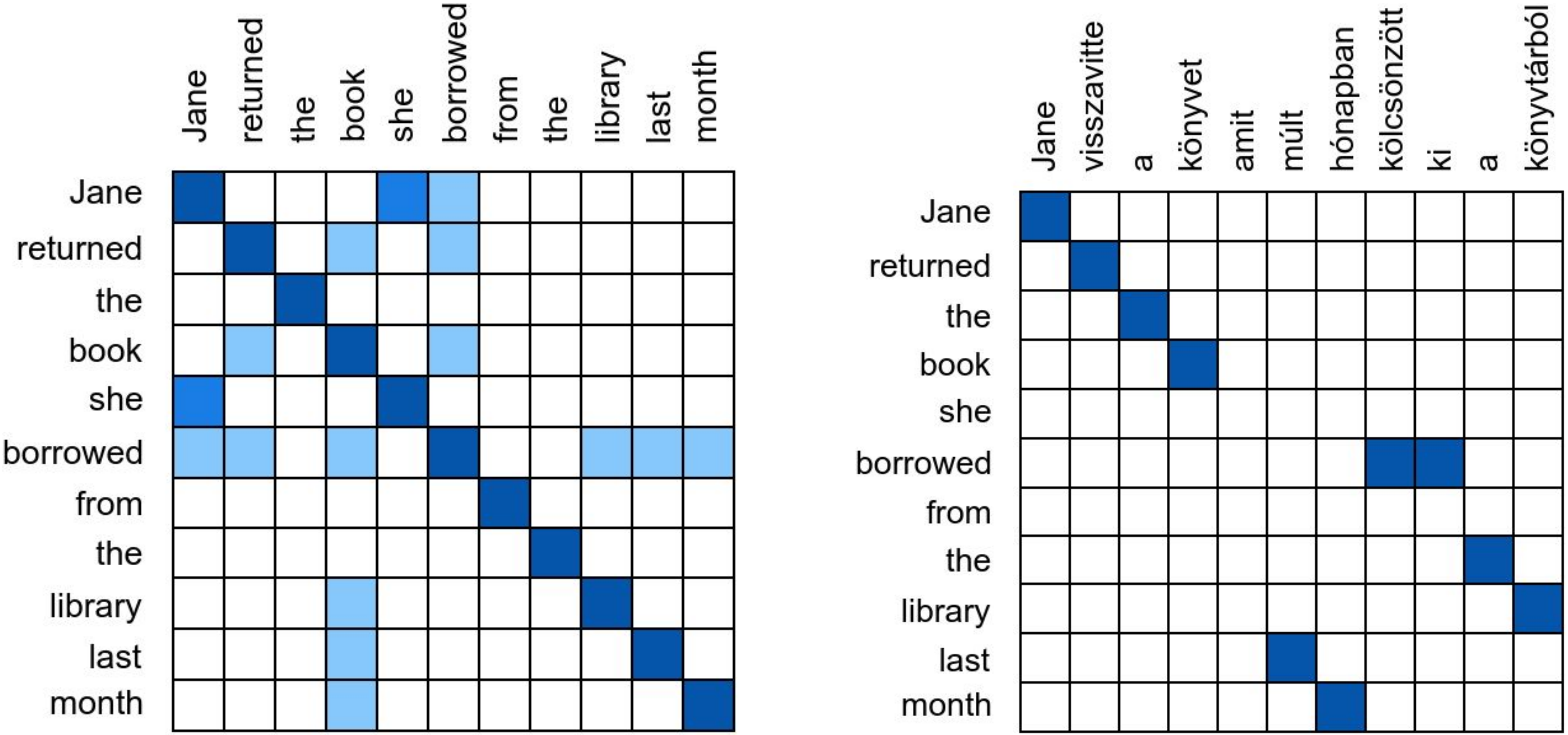


Transformer: Attention, Self-Attention, Cross-Attention

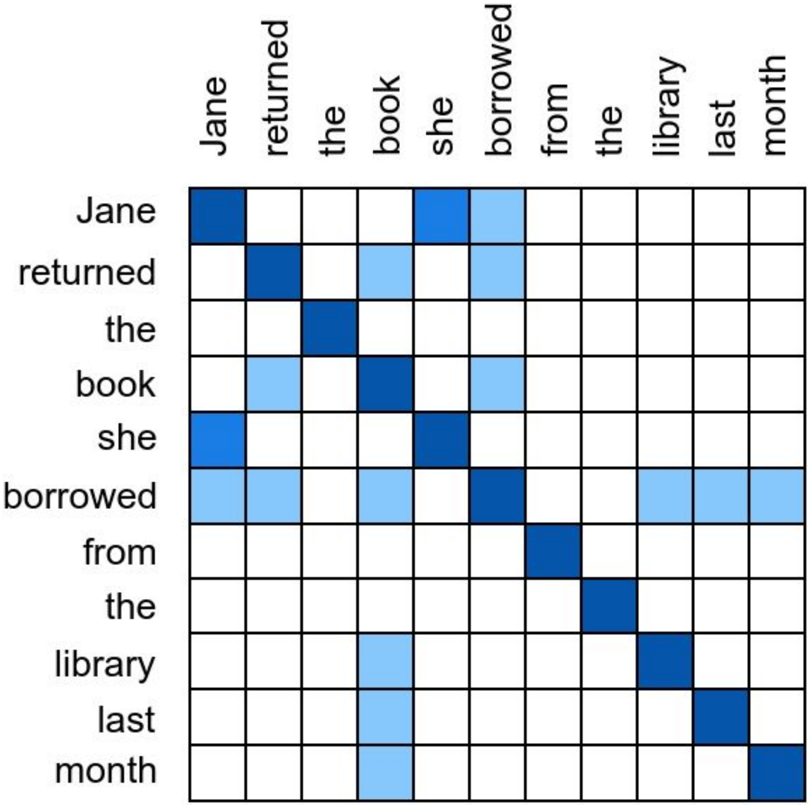


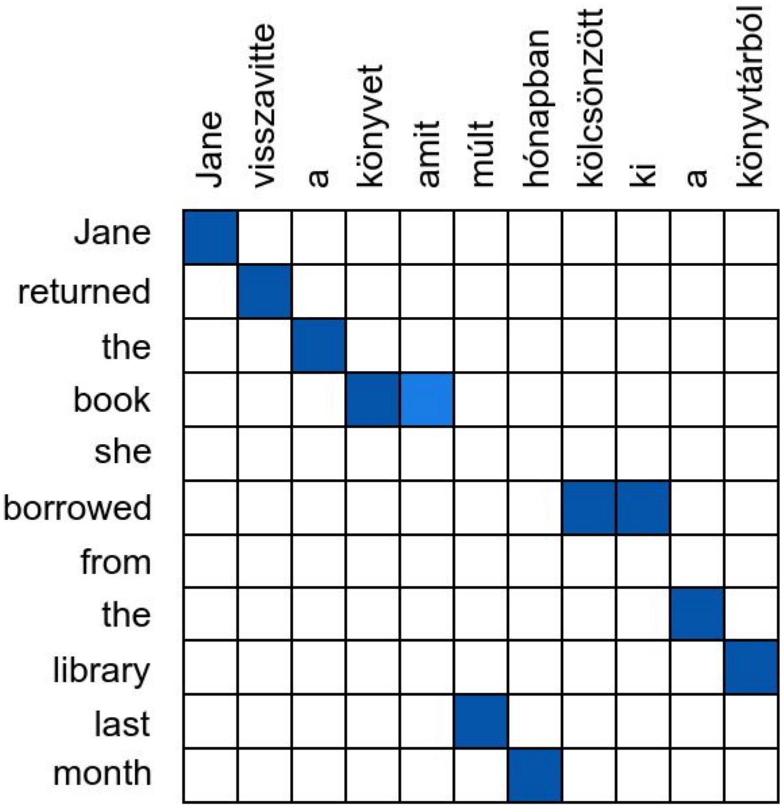


Transformer: Attention, Self-Attention, Cross-Attention

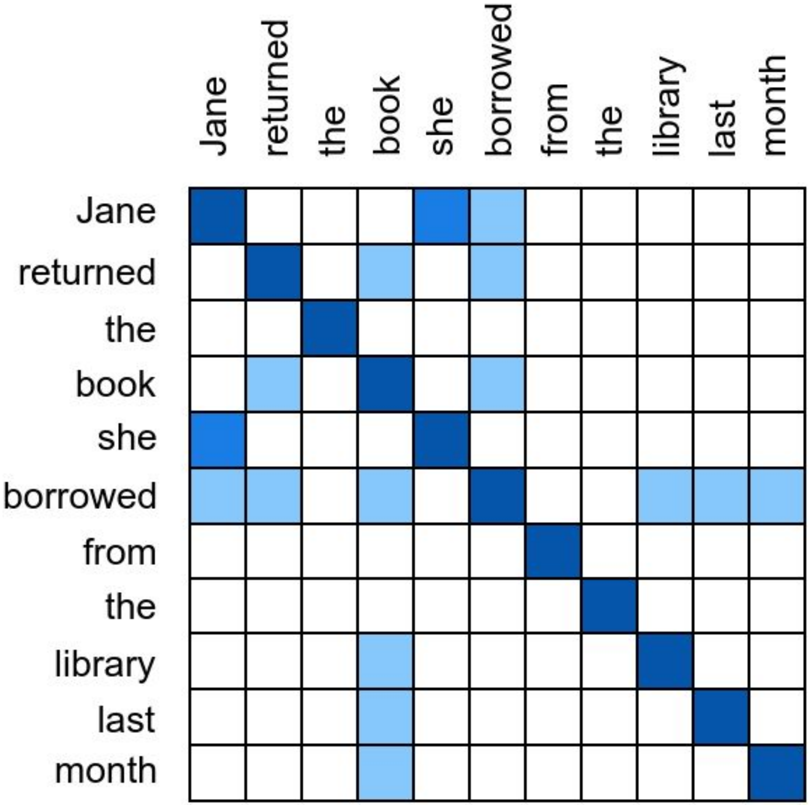


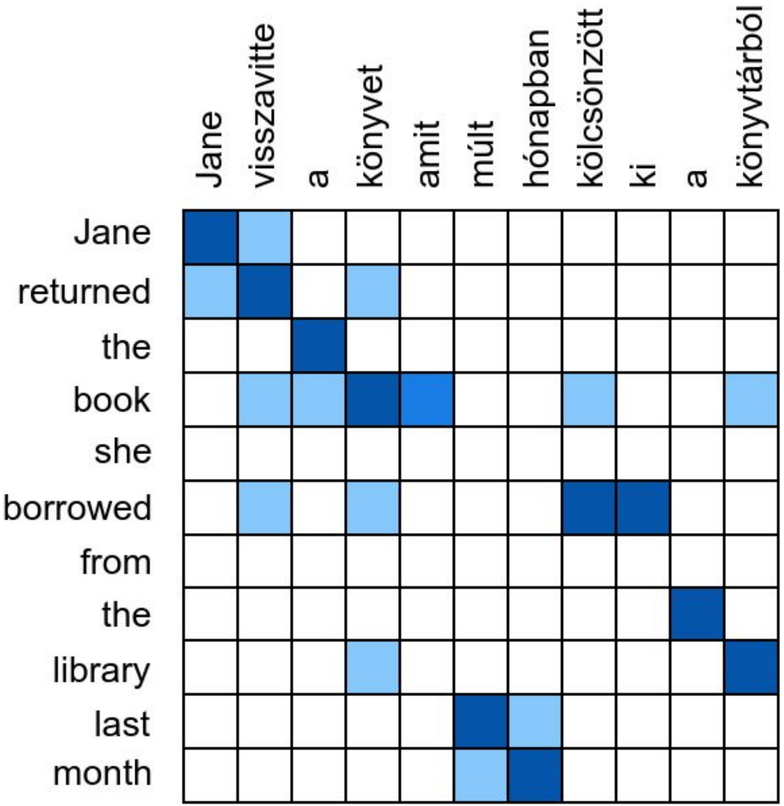
Transformer: Attention, Self-Attention, Cross-Attention



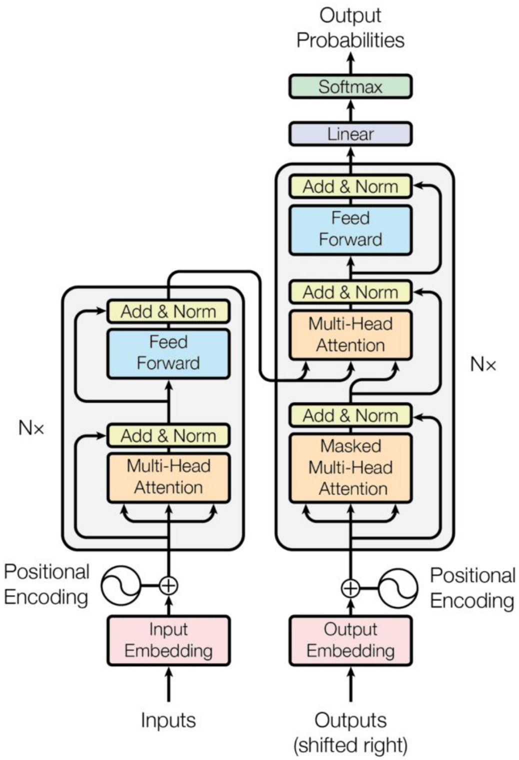


Transformer: Attention, Self-Attention, Cross-Attention



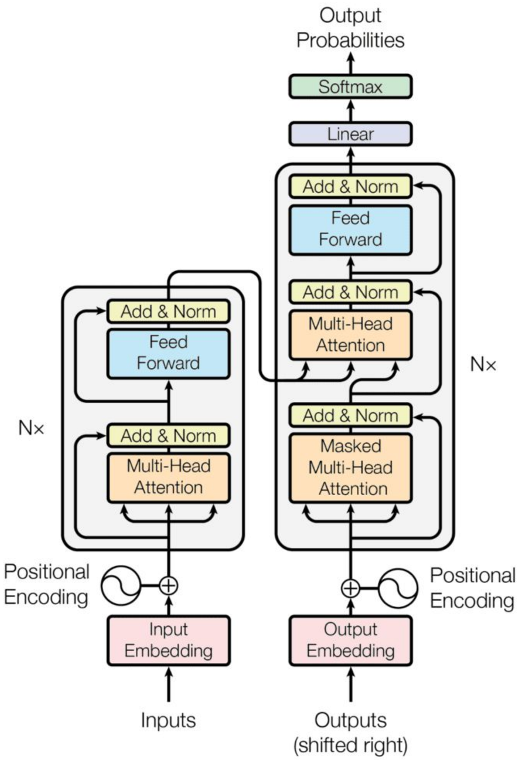


Transformer



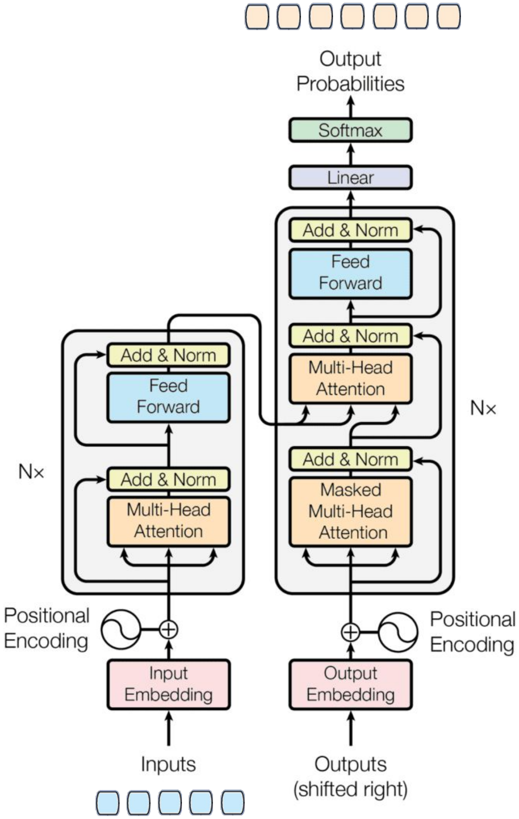
Transformer

Input and Output

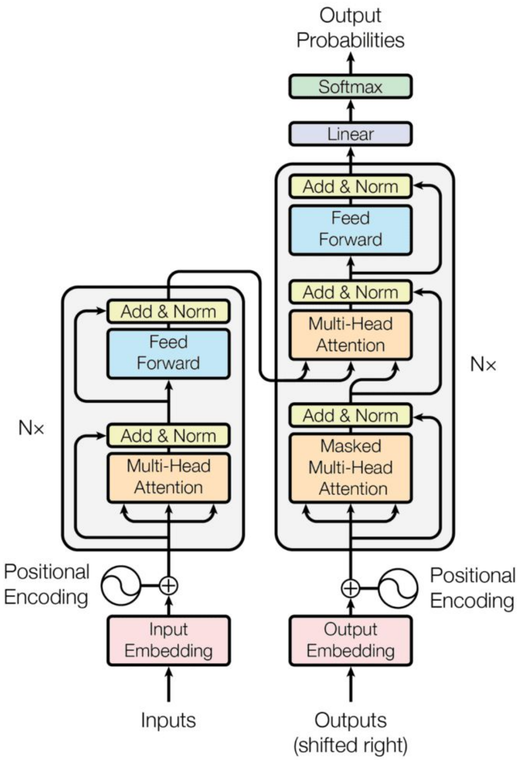


Transformer

Input and Output

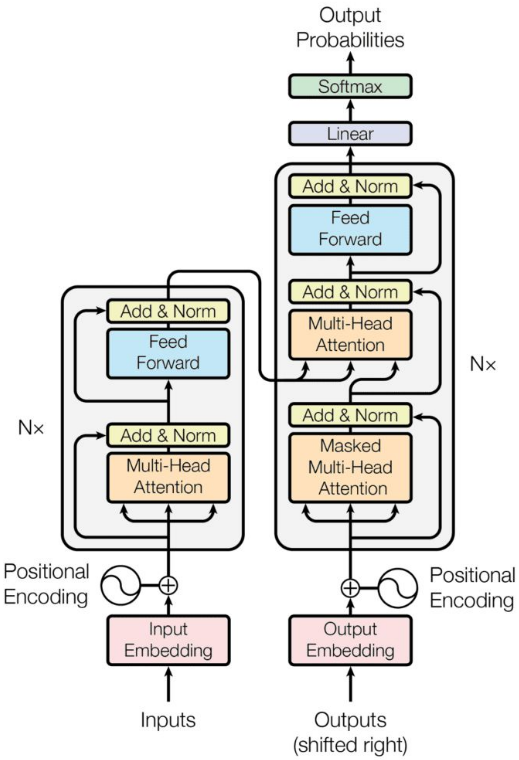


Transformer



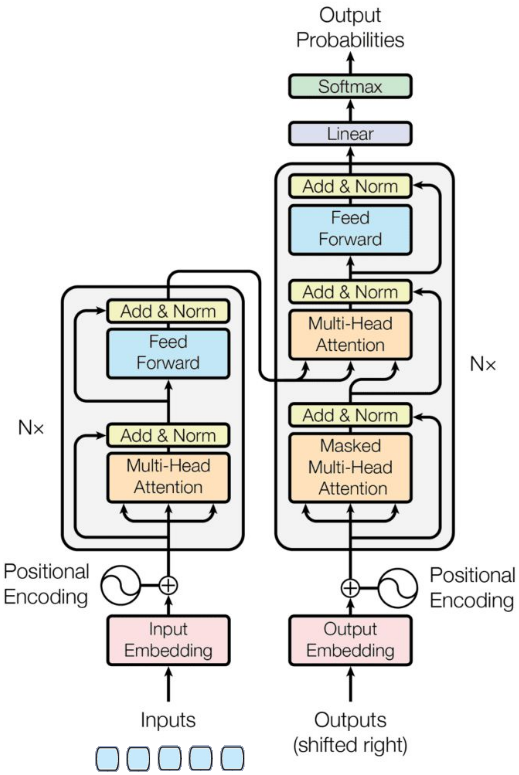
Transformer

Sequential output generation



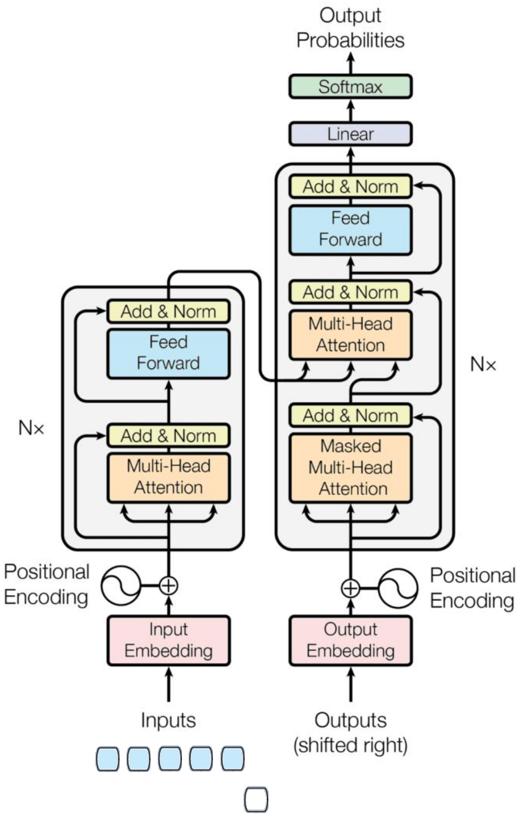
Transformer

Sequential output generation

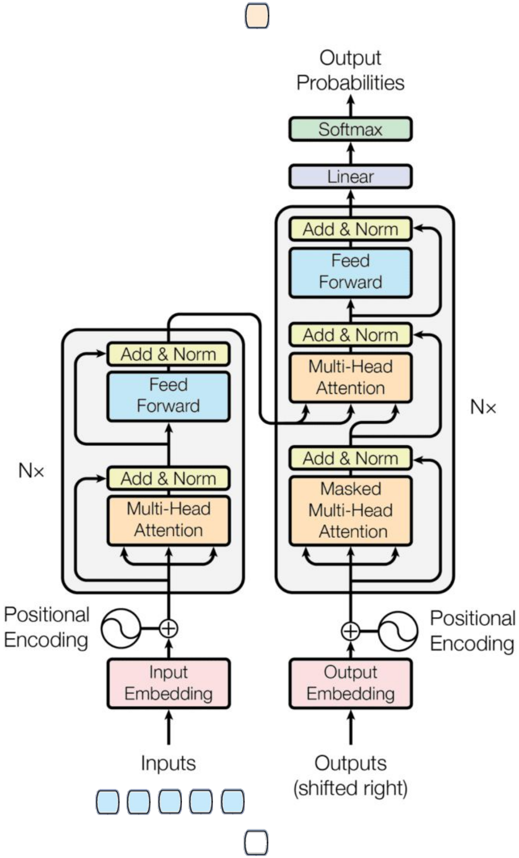


Transformer

Sequential output generation



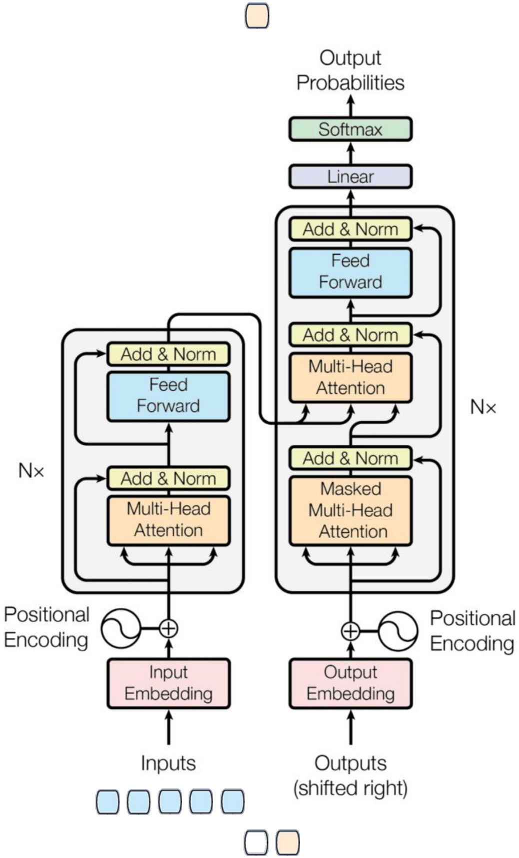
Transformer



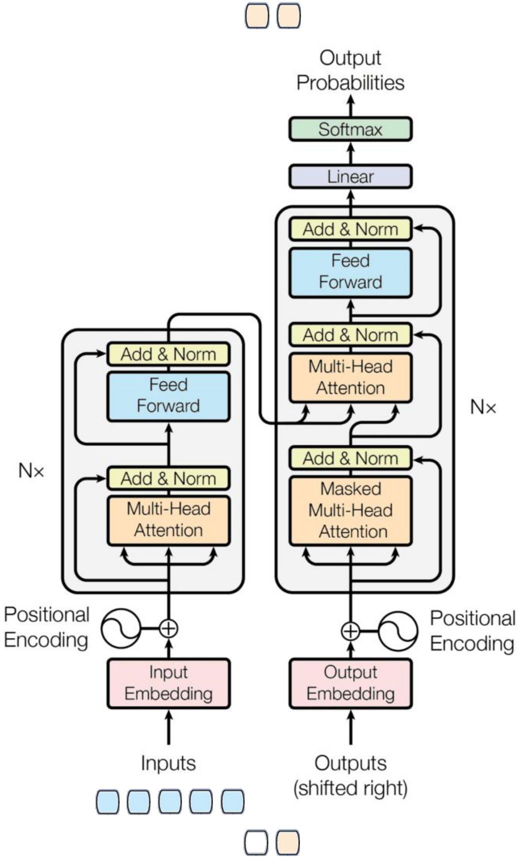
Sequential output generation

Transformer

Sequential output generation

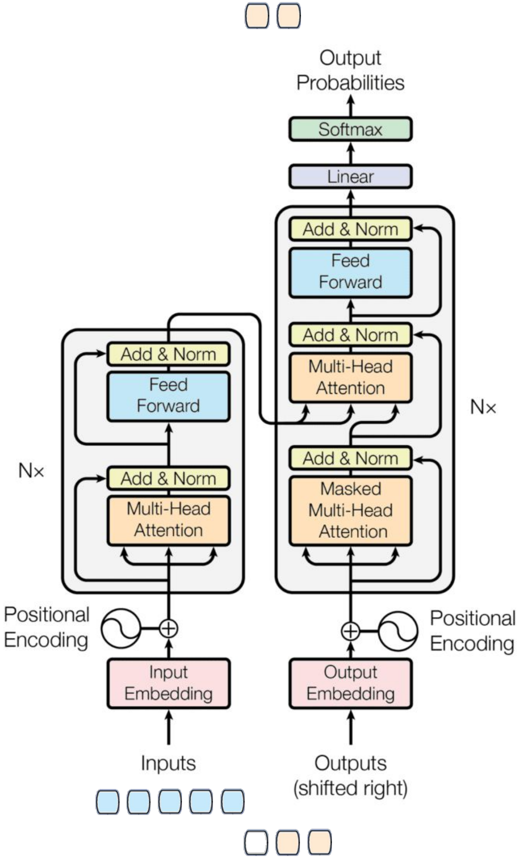


Transformer



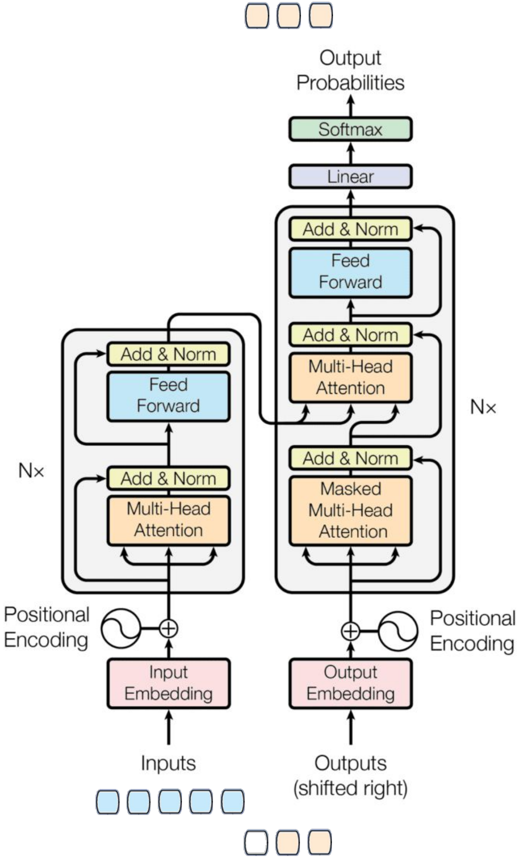
Sequential output generation

Transformer



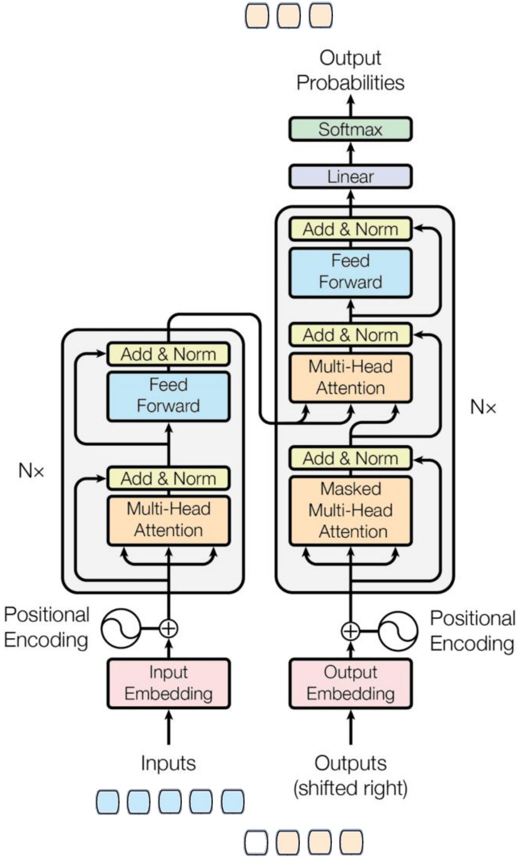
Sequential output generation

Transformer



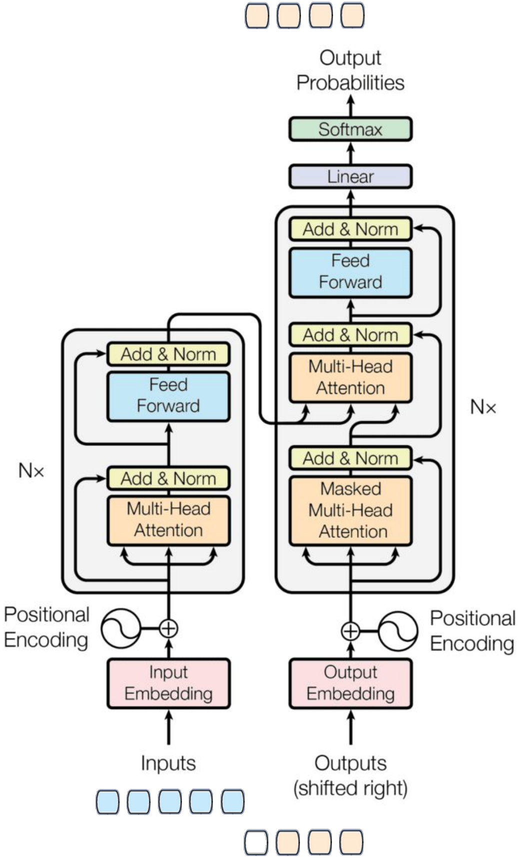
Sequential output generation

Transformer



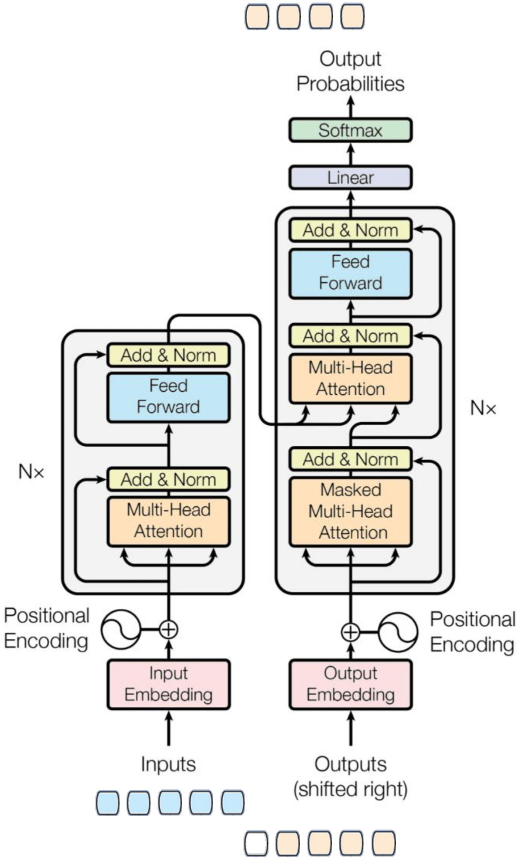
Sequential output generation

Transformer



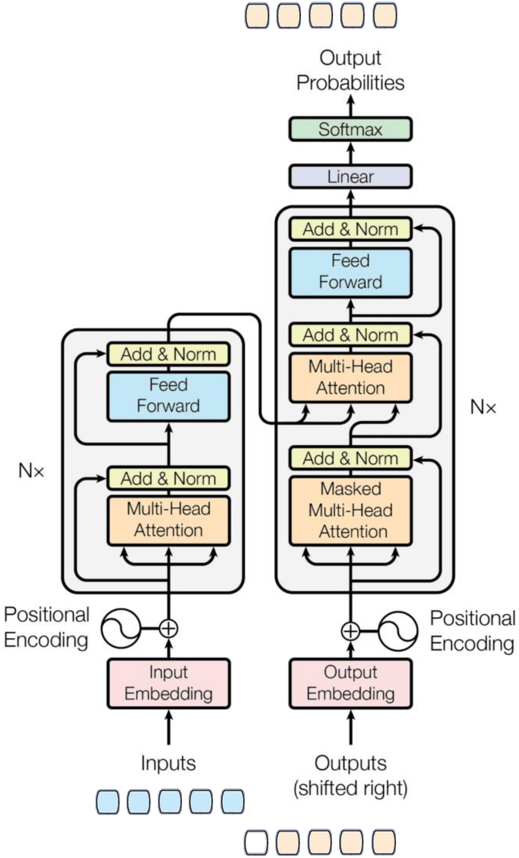
Sequential output generation

Transformer



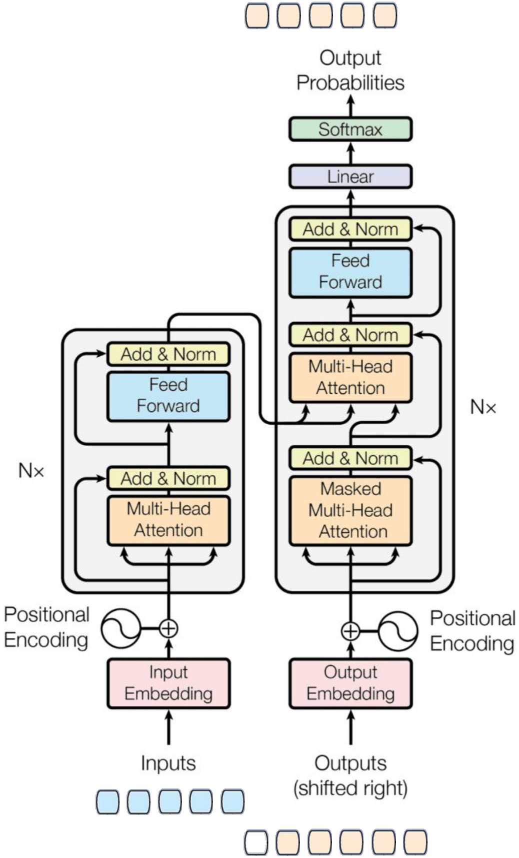
Sequential output generation

Transformer



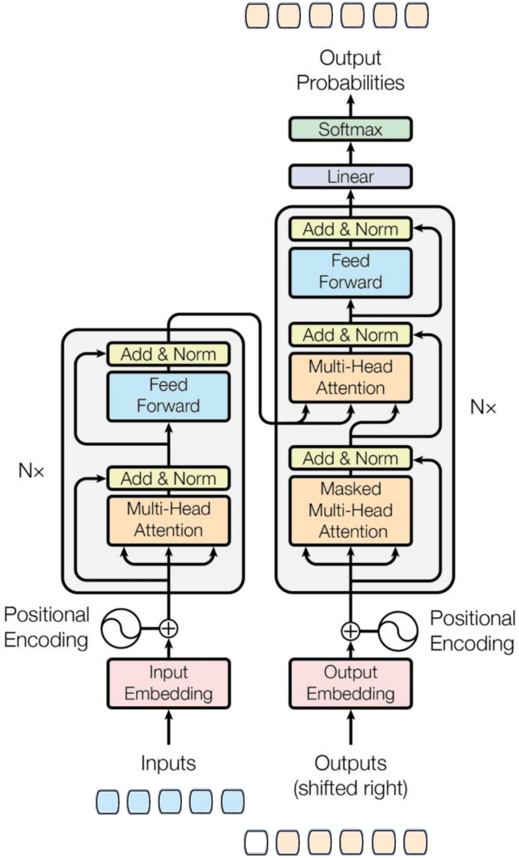
Sequential output generation

Transformer



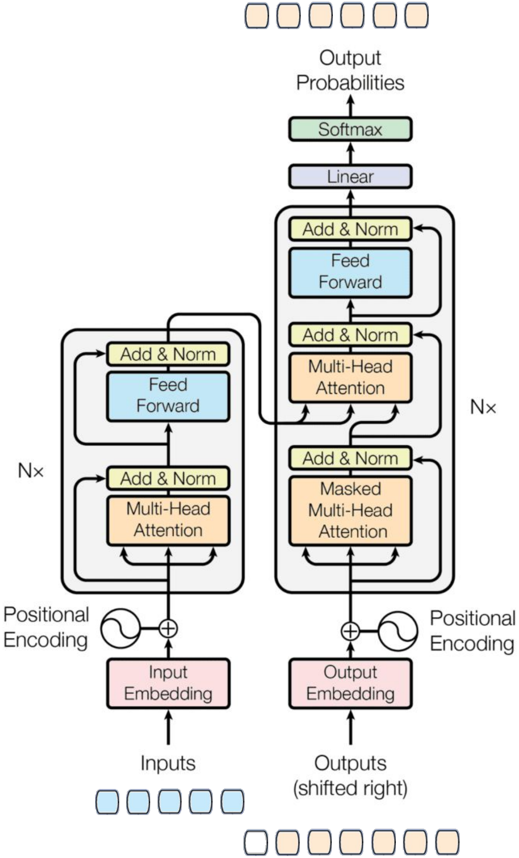
Sequential output generation

Transformer



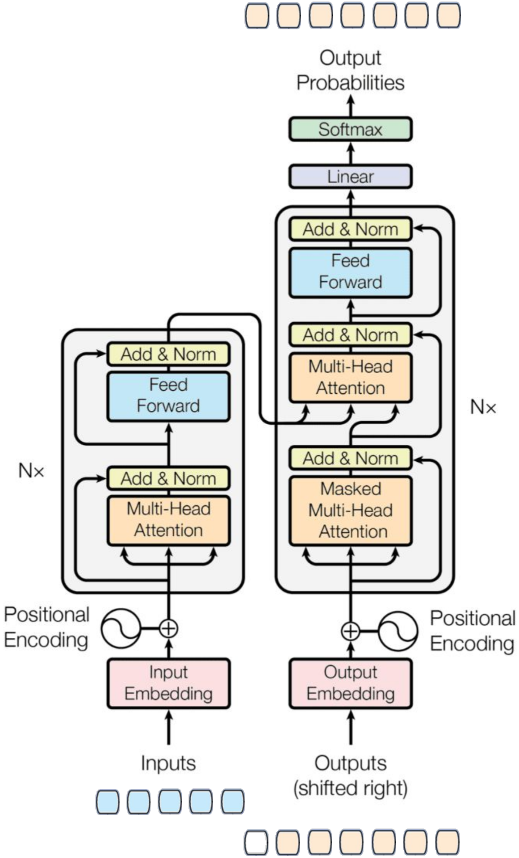
Sequential output generation

Transformer



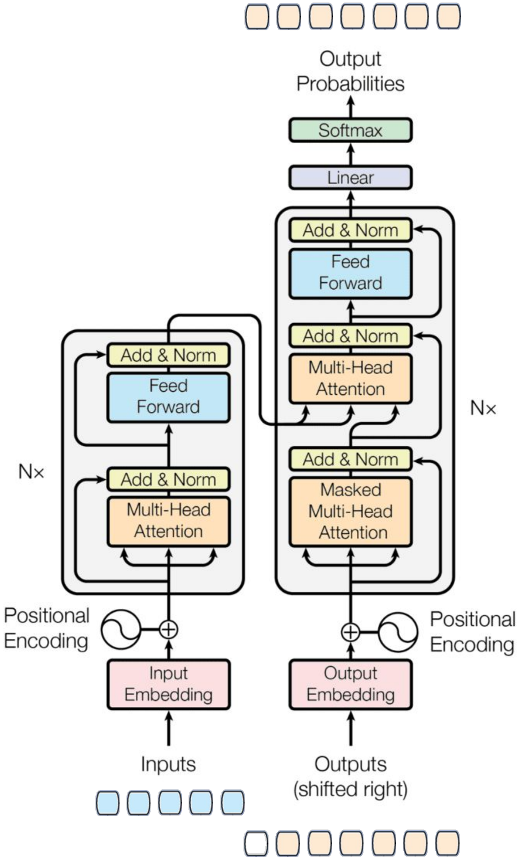
Sequential output generation

Transformer



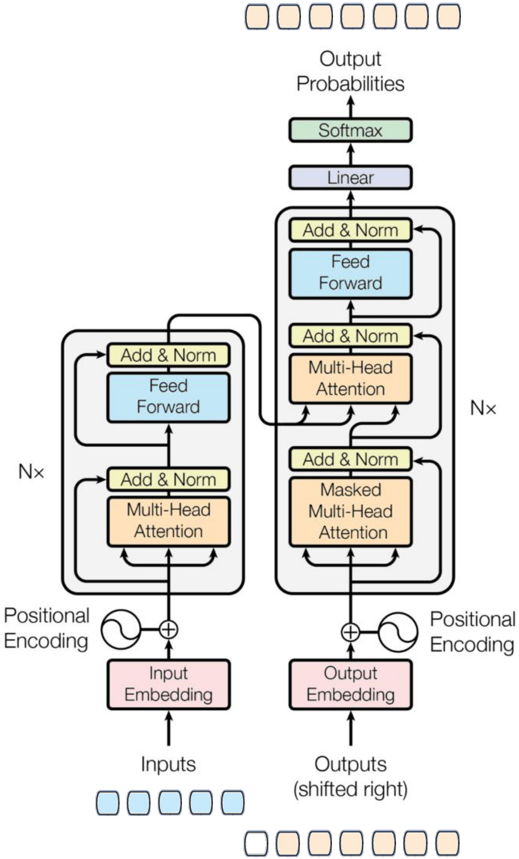
Sequential output generation

Transformer

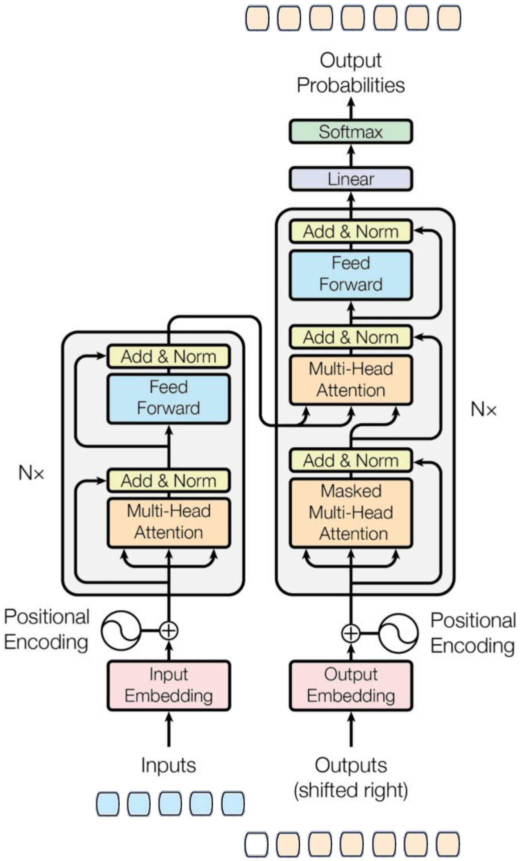


Sequential output generation

Transformer



Transformer



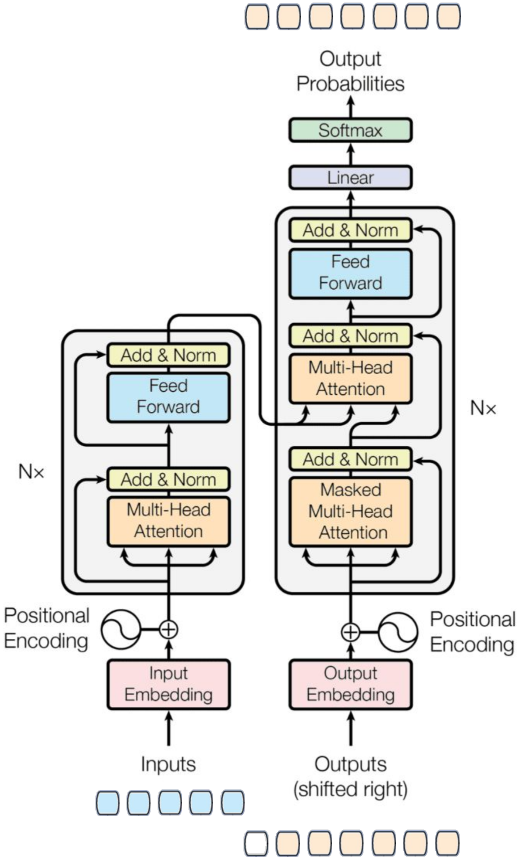
Training time:

- parallel

Inference time:

- sequential

Transformer



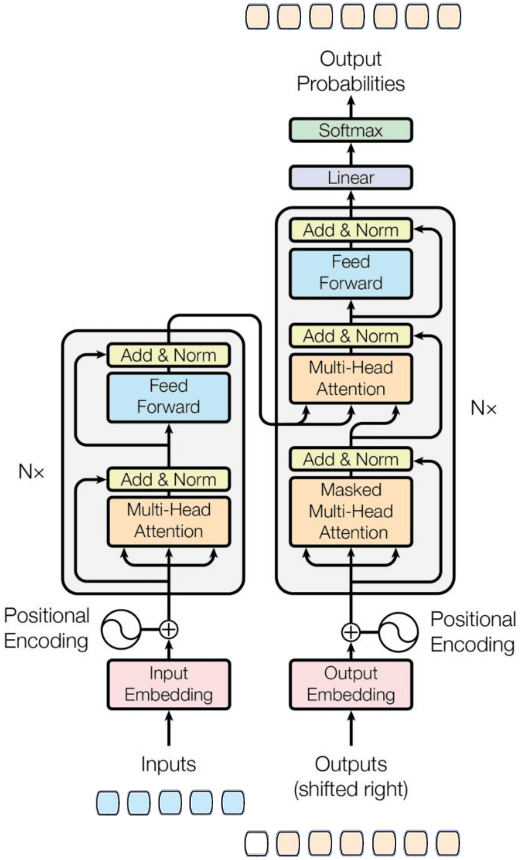
Training time:

- parallel

Inference time:

- sequential

Transformer



Masking:

- avoid peeking

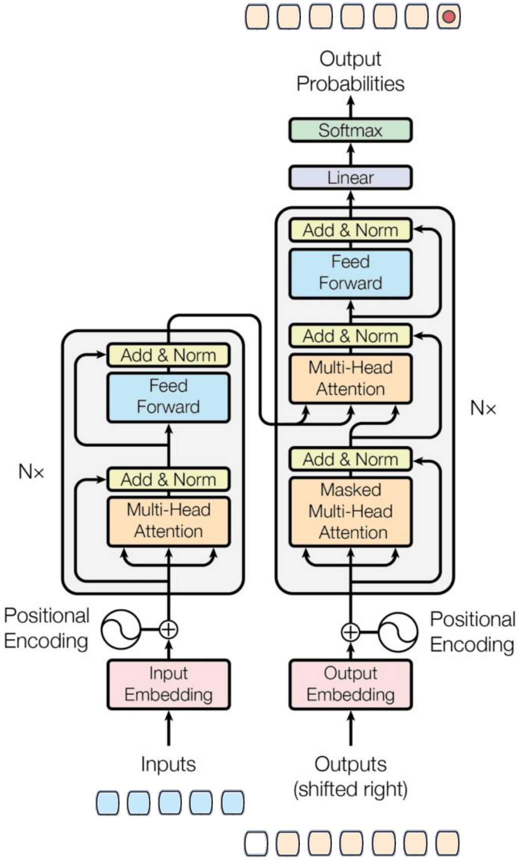
Training time:

- parallel

Inference time:

- sequential

Transformer



Masking:

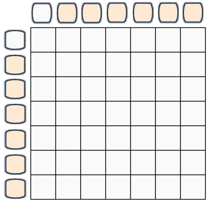
- avoid peeking

Training time:

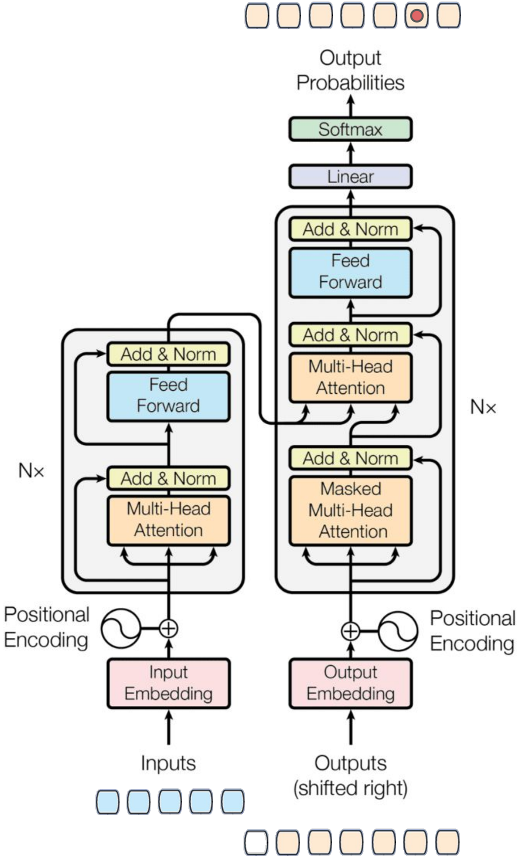
- parallel

Inference time:

- sequential



Transformer



Masking:

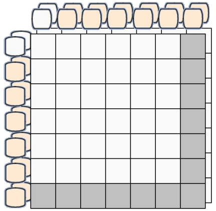
- avoid peeking

Training time:

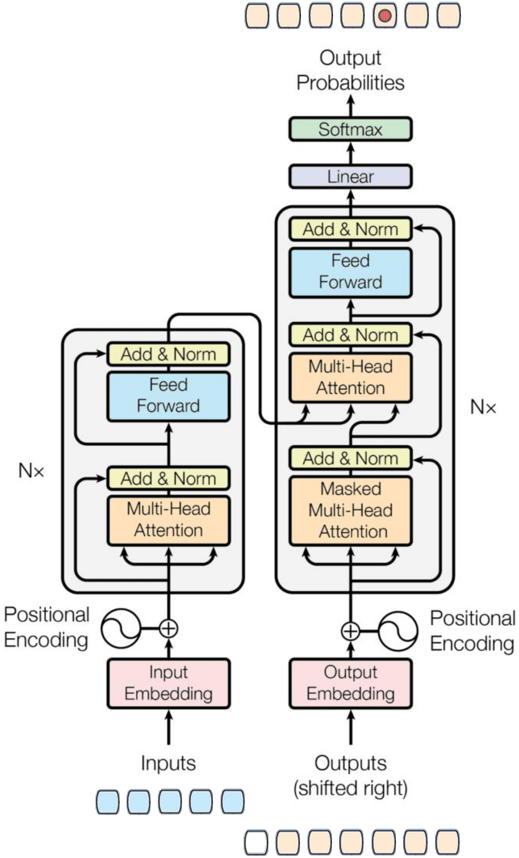
- parallel

Inference time:

- sequential



Transformer



Masking:

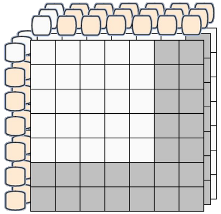
- avoid peeking

Training time:

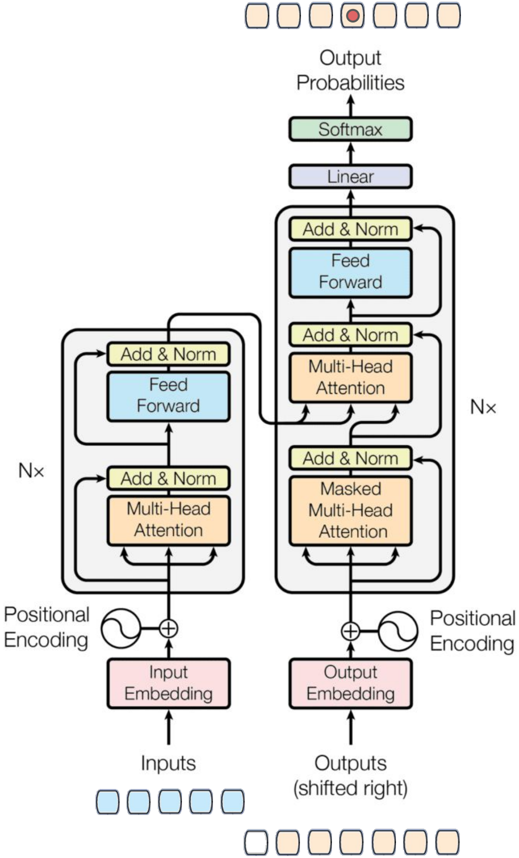
- parallel

Inference time:

- sequential



Transformer



Masking:

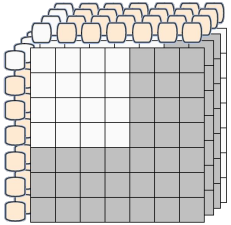
- avoid peeking

Training time:

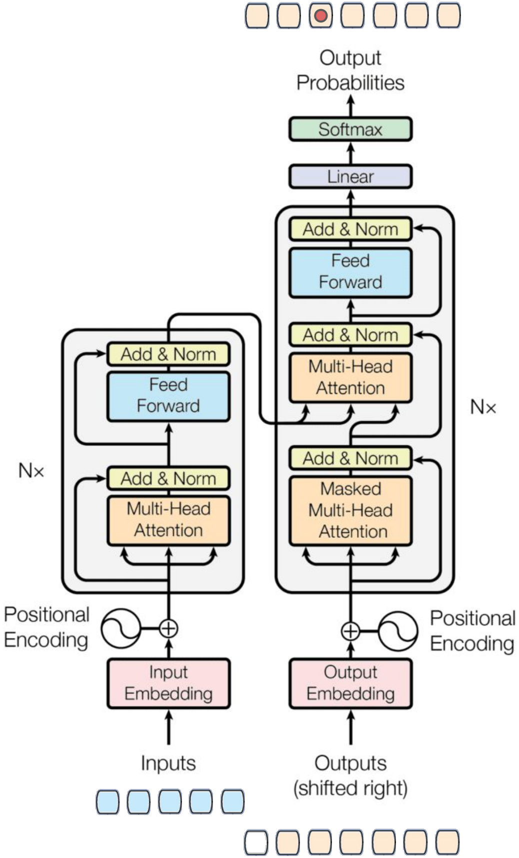
- parallel

Inference time:

- sequential



Transformer



Masking:

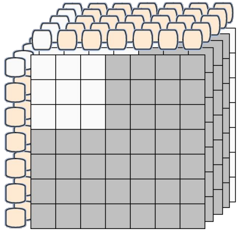
- avoid peeking

Training time:

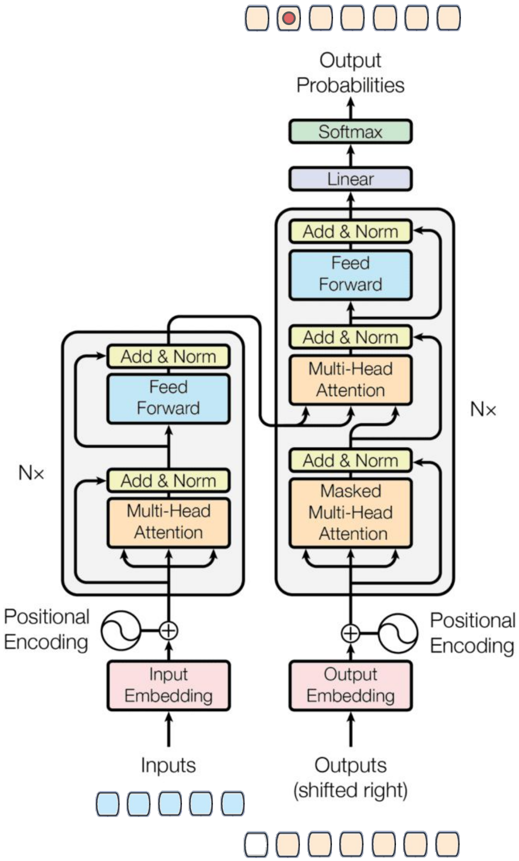
- parallel

Inference time:

- sequential



Transformer



Masking:

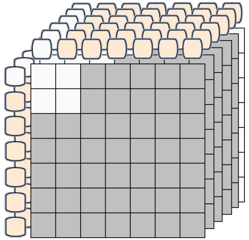
- avoid peeking

Training time:

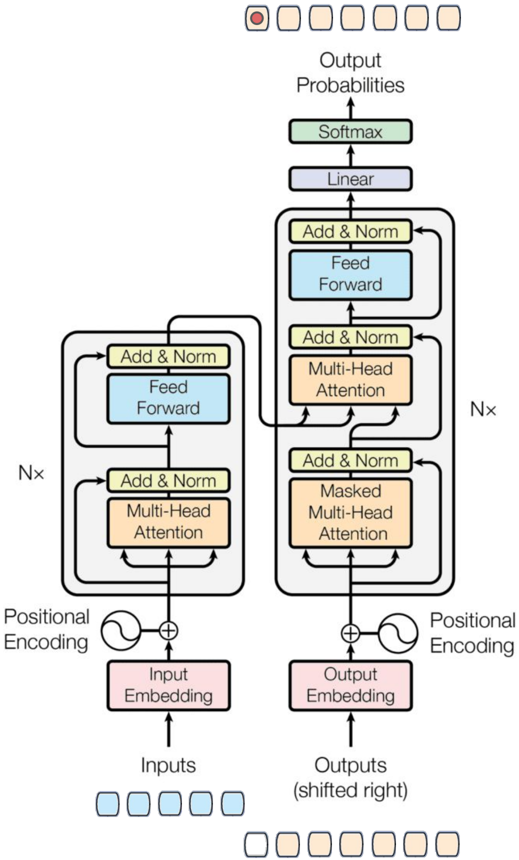
- parallel

Inference time:

- sequential



Transformer



Masking:

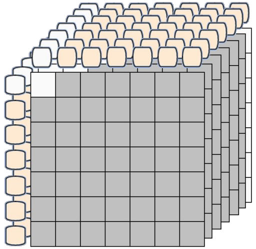
- avoid peeking

Training time:

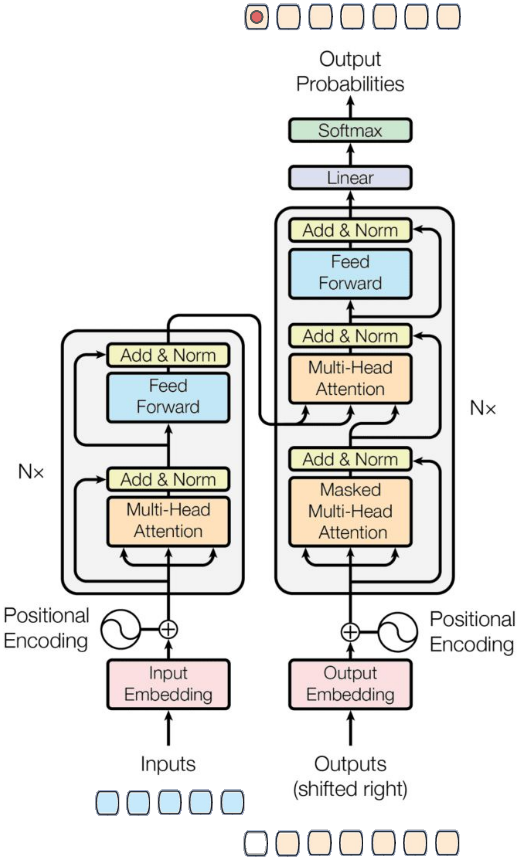
- parallel

Inference time:

- sequential



Transformer



Masking:

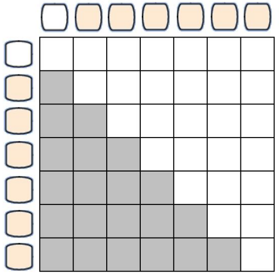
- avoid peeking

Training time:

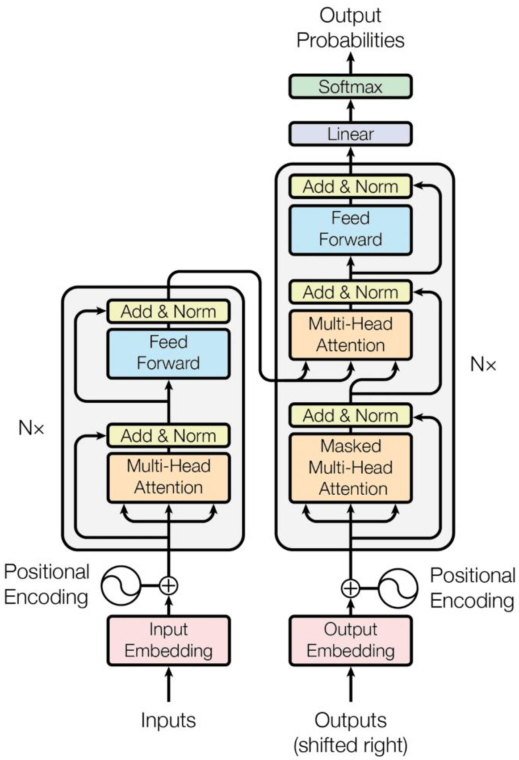
- parallel

Inference time:

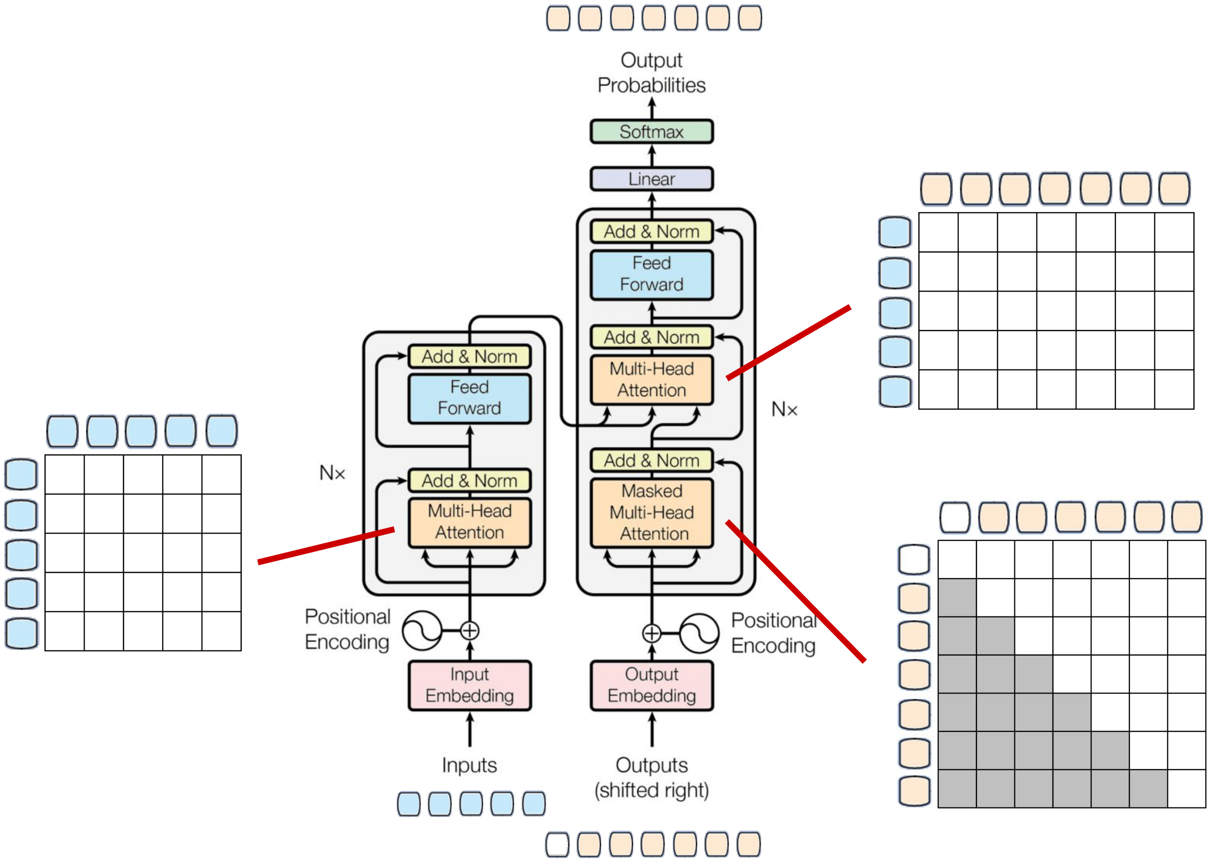
- sequential



Transformer



Transformer

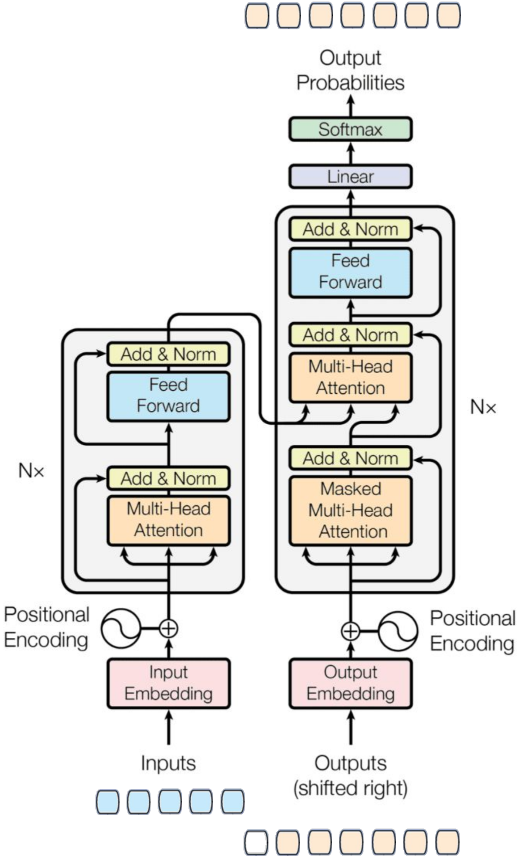


Self-Attention

Cross-Attention

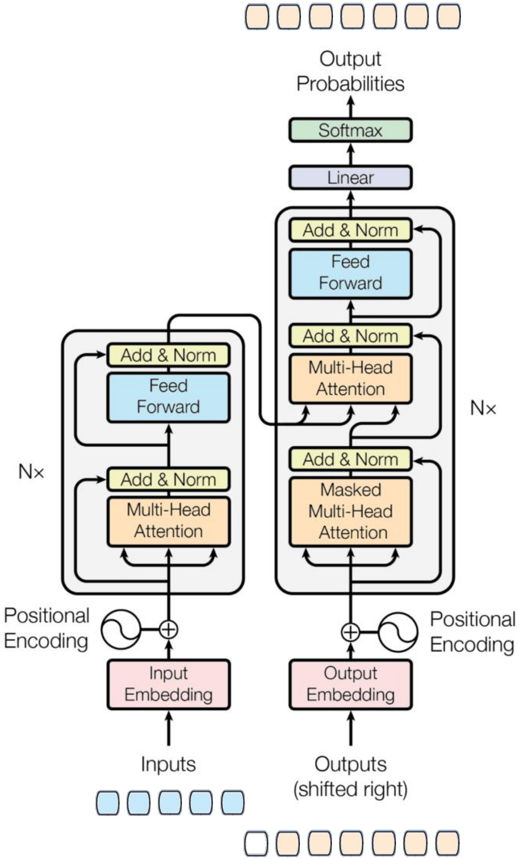
Masked
Self-Attention

Transformer



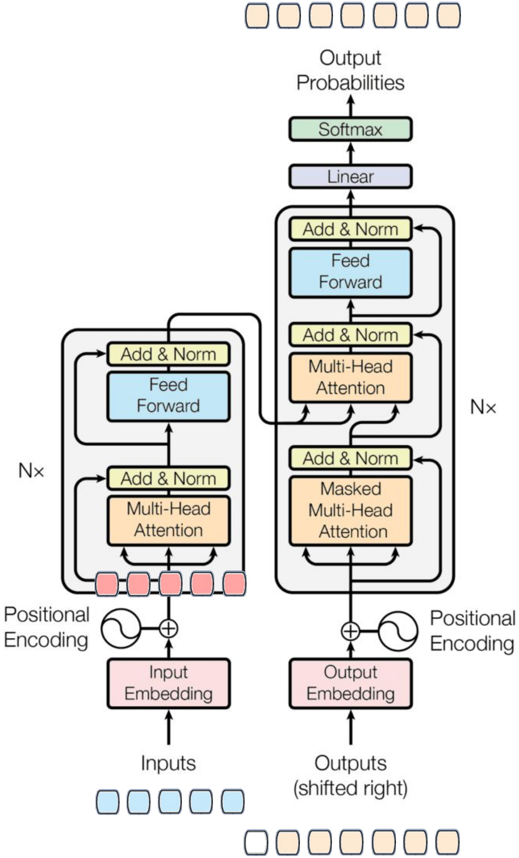
Transformer

Hidden representations



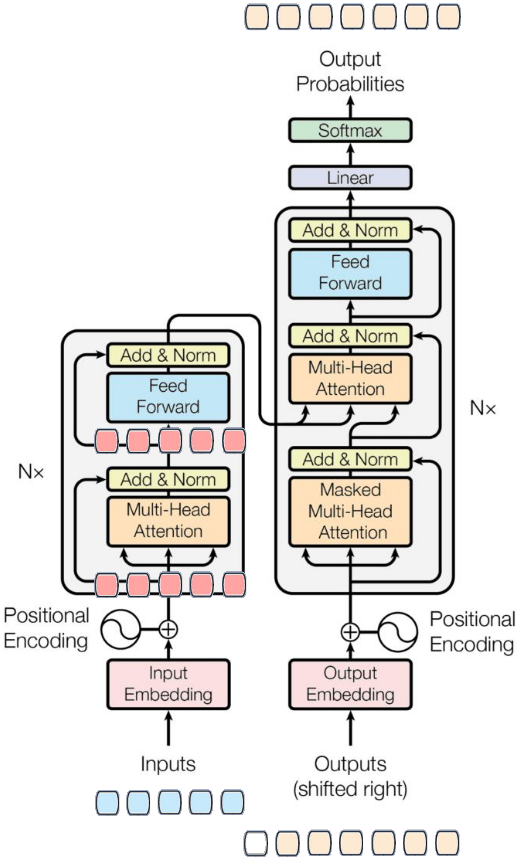
Transformer

Hidden representations



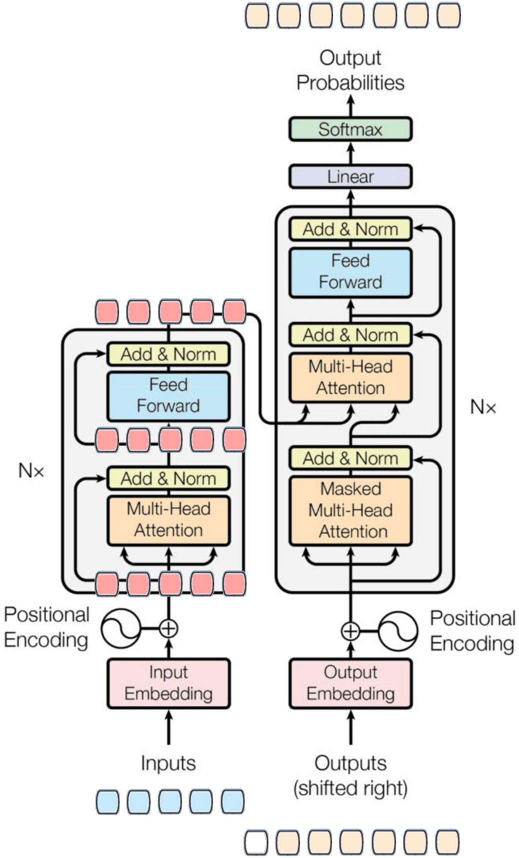
Transformer

Hidden representations



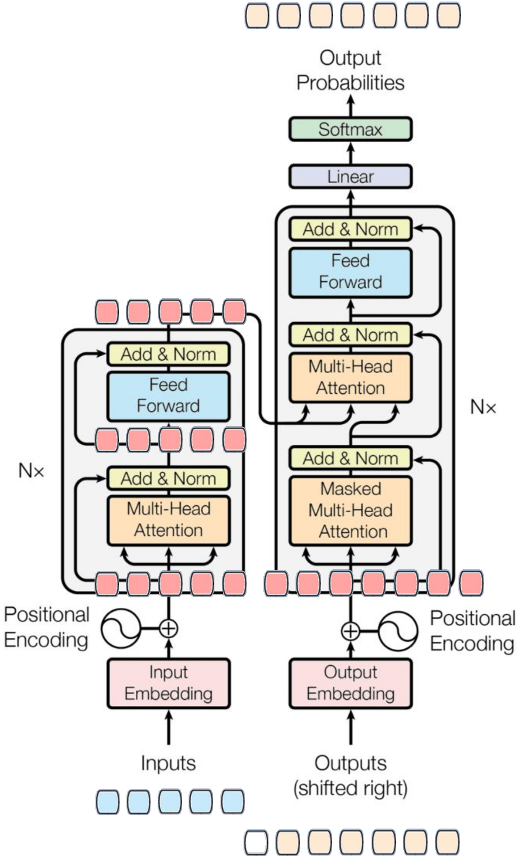
Transformer

Hidden representations



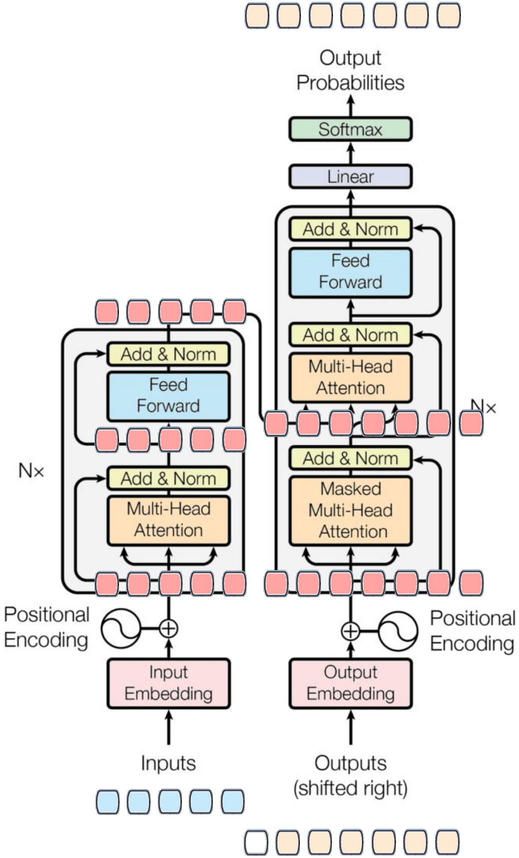
Transformer

Hidden representations



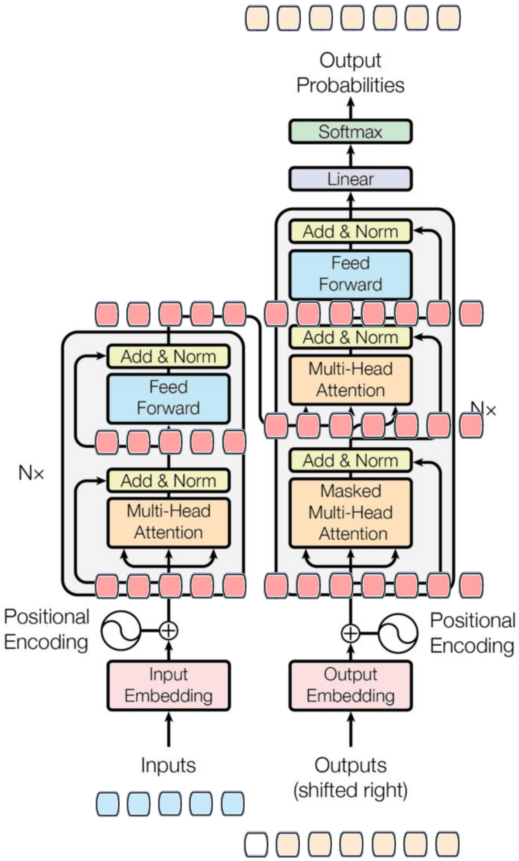
Transformer

Hidden representations



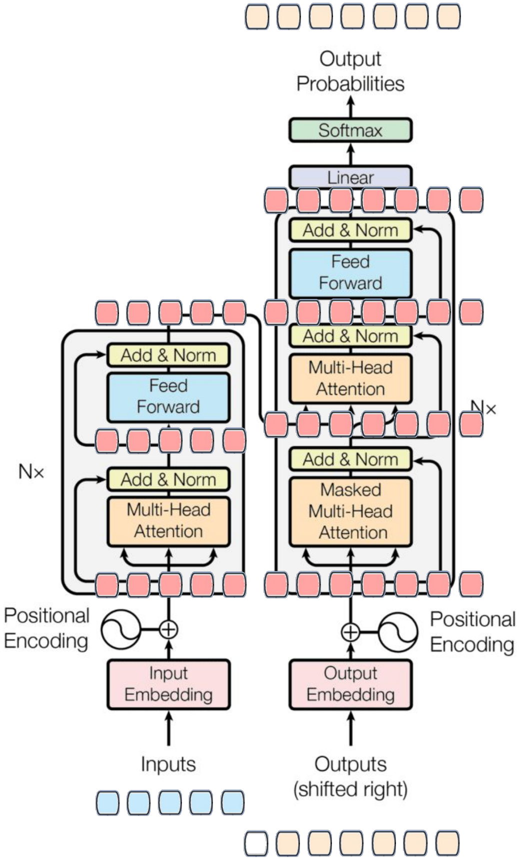
Transformer

Hidden representations



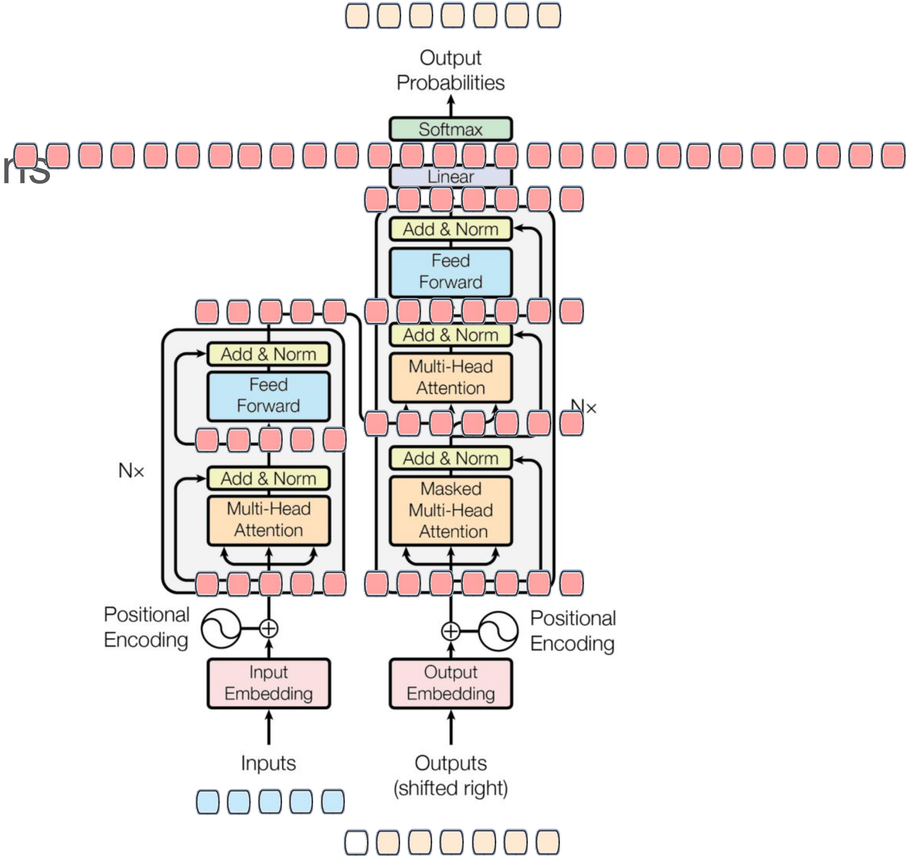
Transformer

Hidden representations



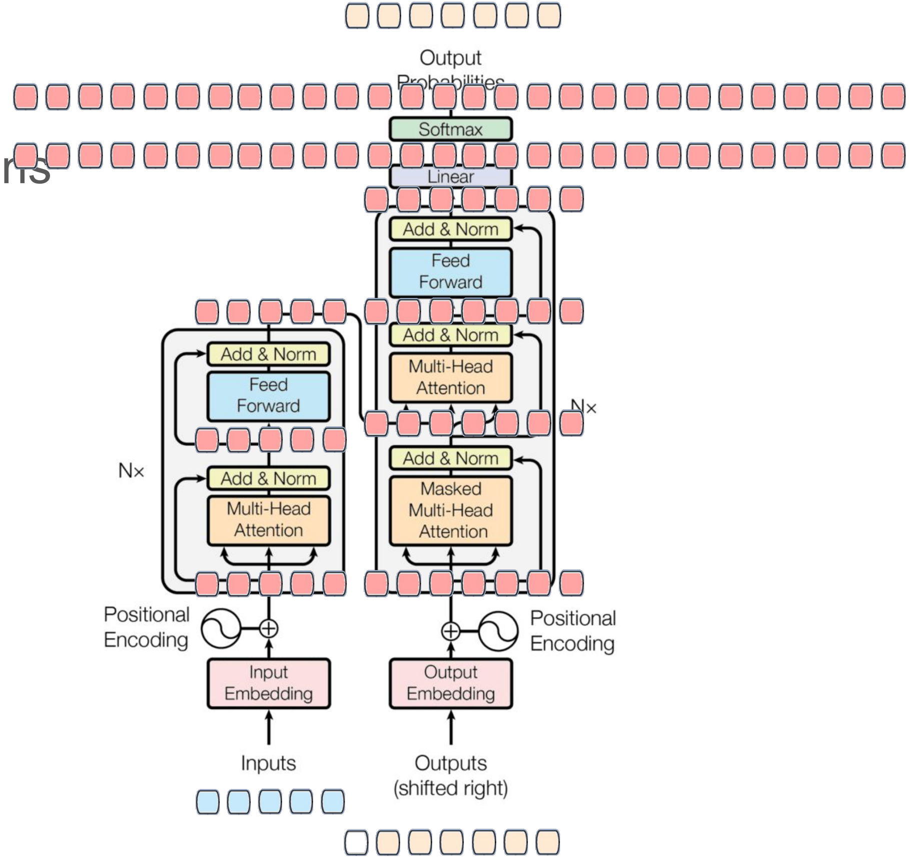
Transformer

Hidden representations



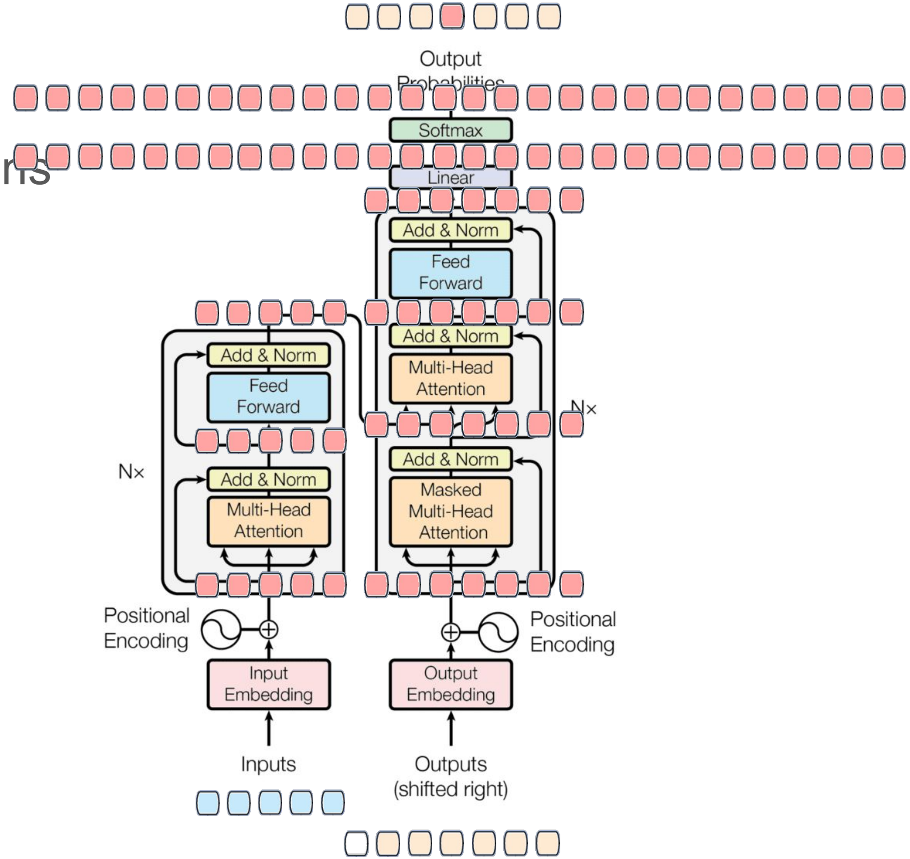
Transformer

Hidden representations

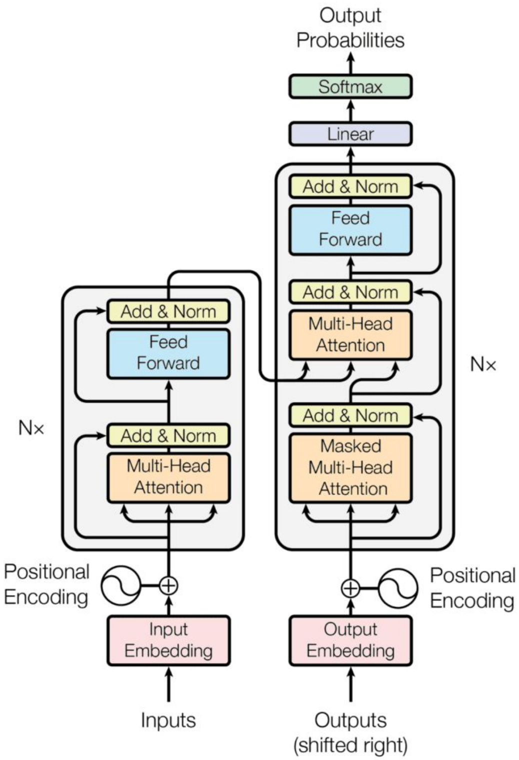


Transformer

Hidden representations

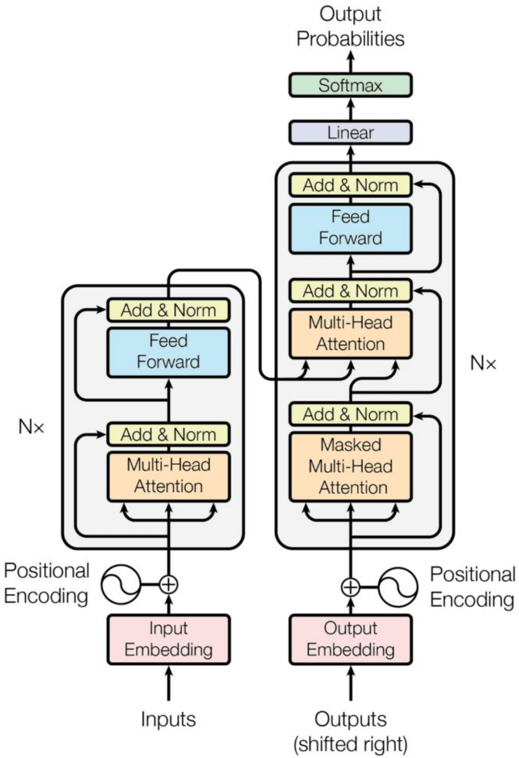


Transformer

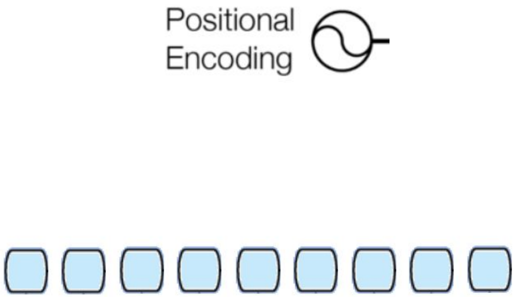


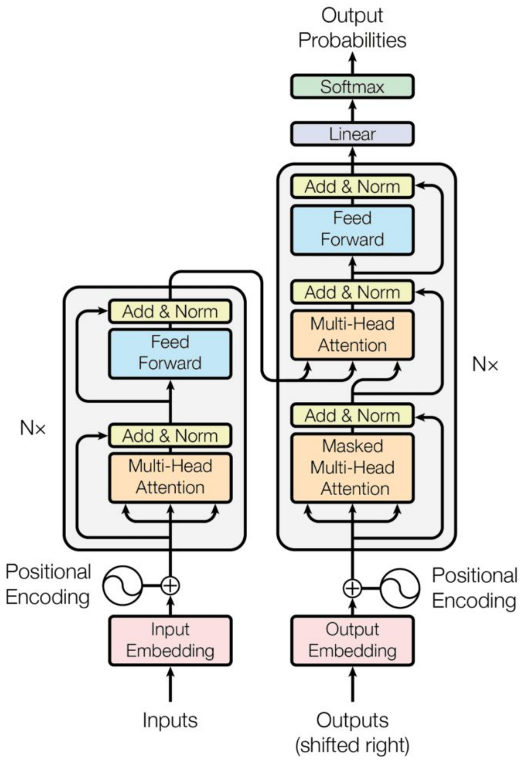
Transformer

Positional
Encoding

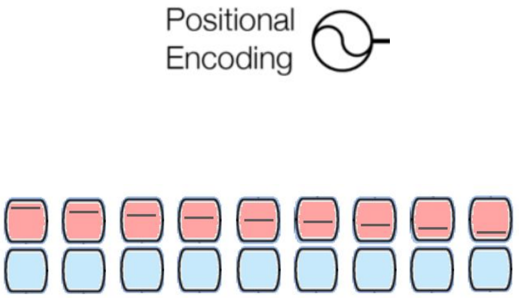


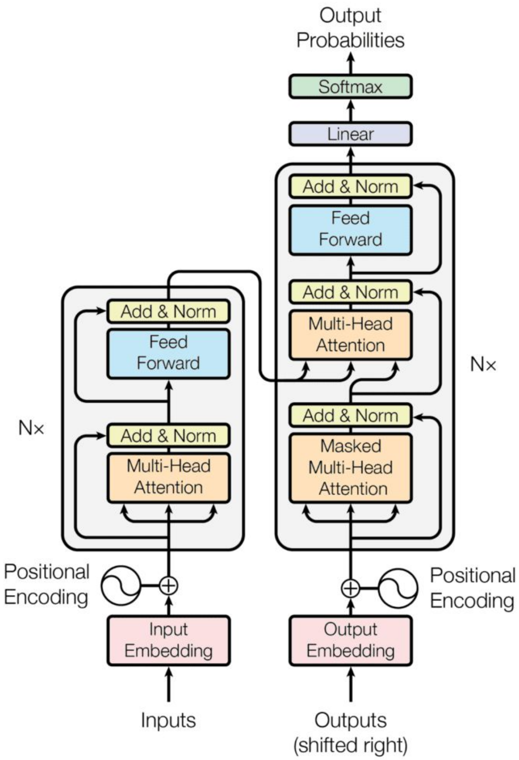
Transformer



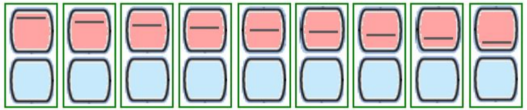


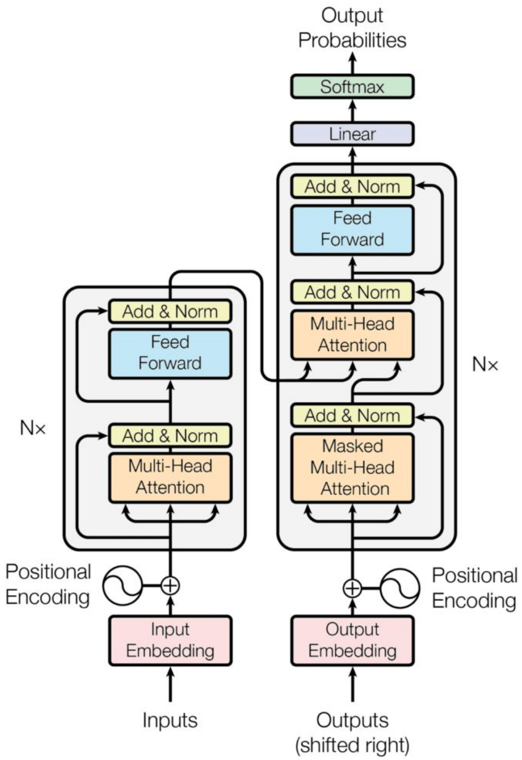
Transformer



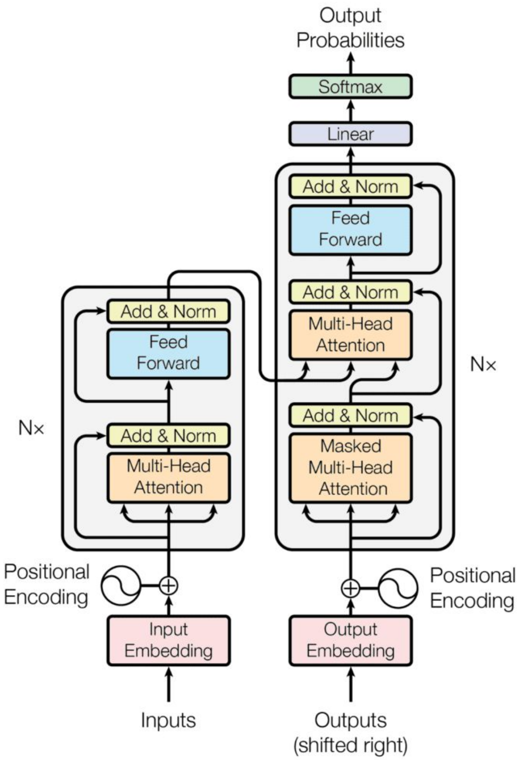


Transformer



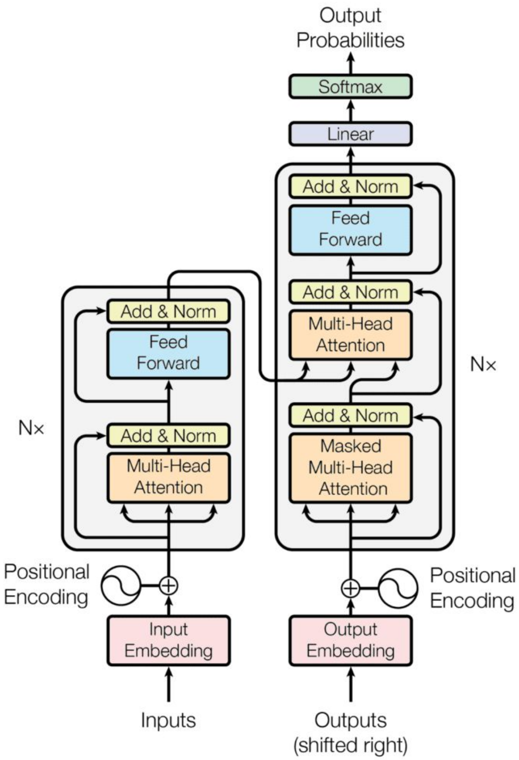


Transformer

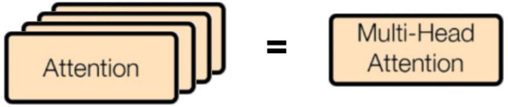


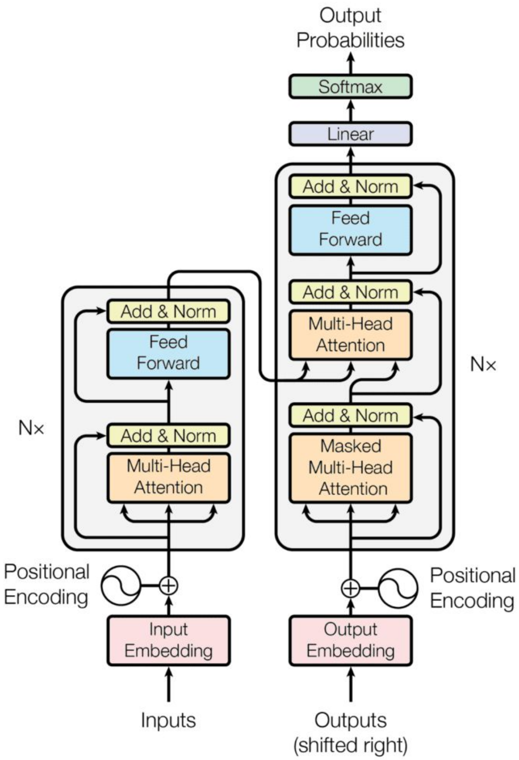
Transformer

Multi-Head Attention

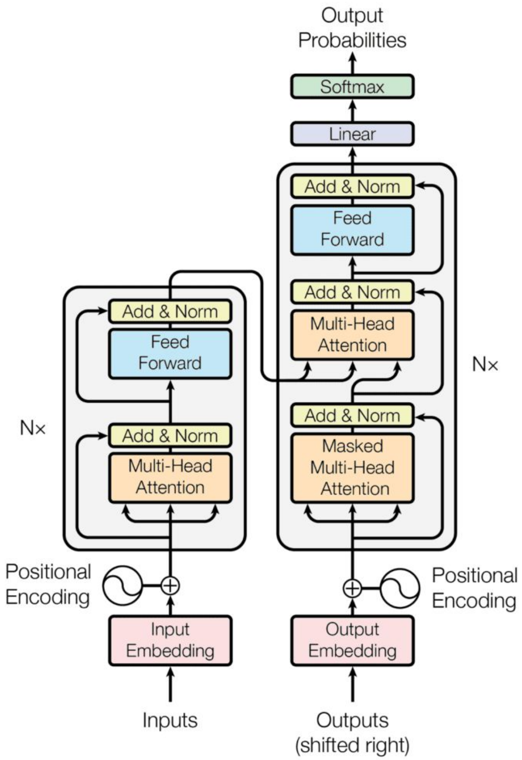


Transformer

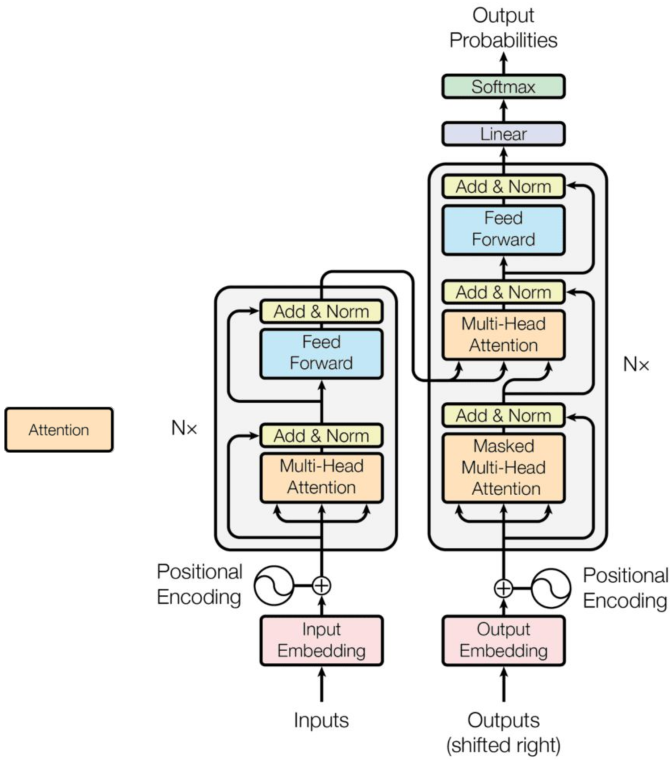




Transformer



Transformer

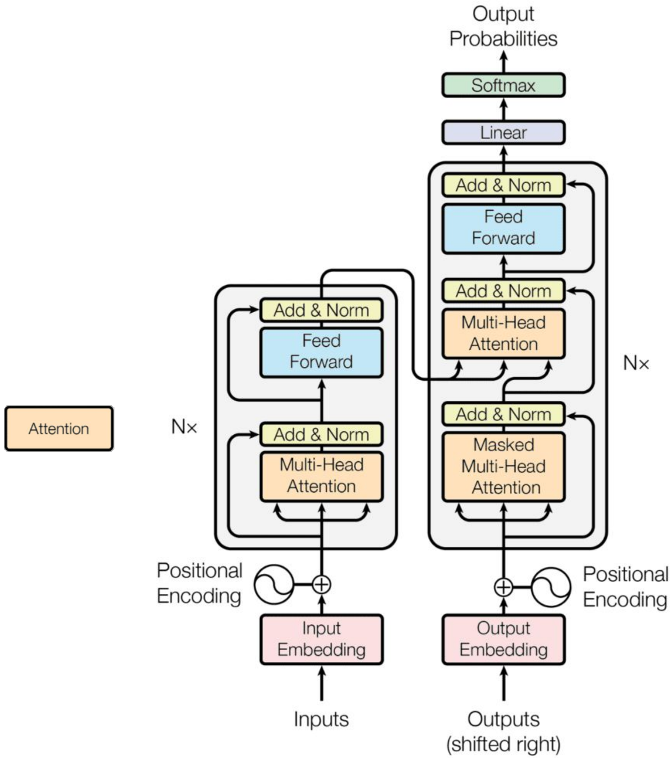


Transformer: Attention - Query, Key, Value

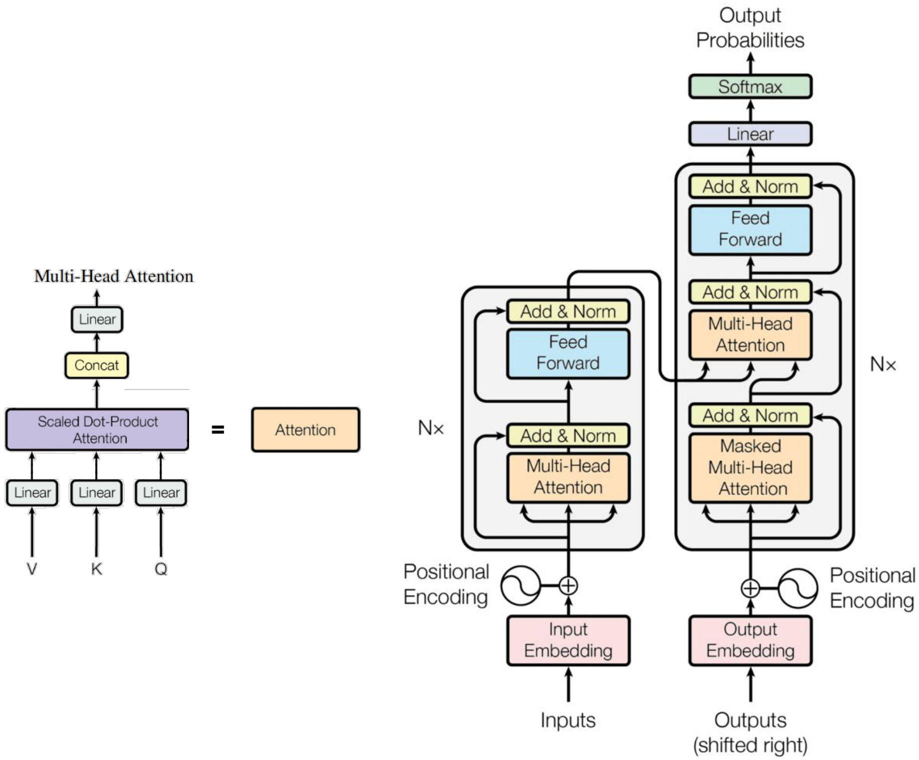
Examples:

- Recruition
- Article searching
- Online shopping

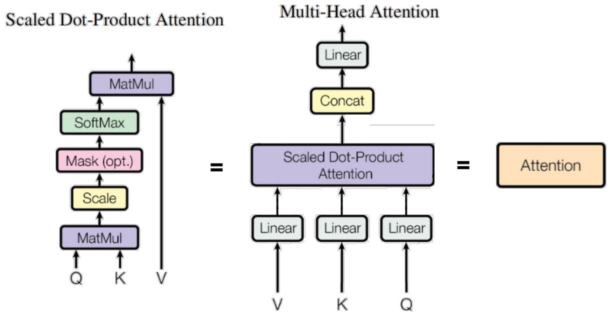
Transformer

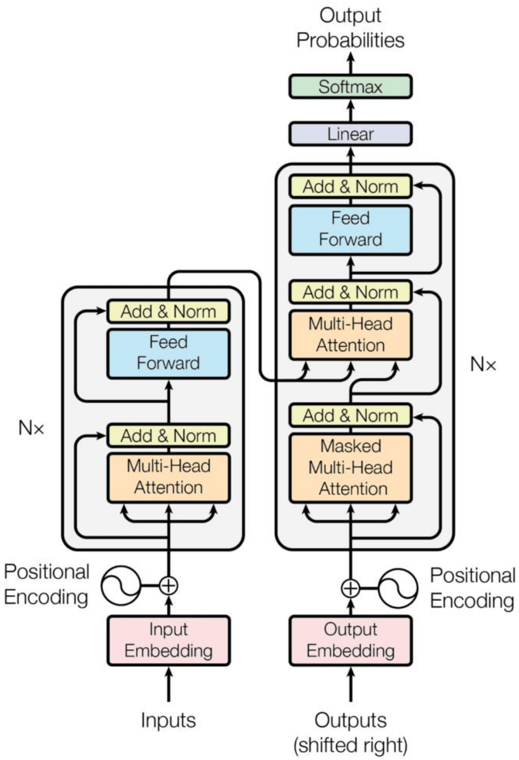


Transformer

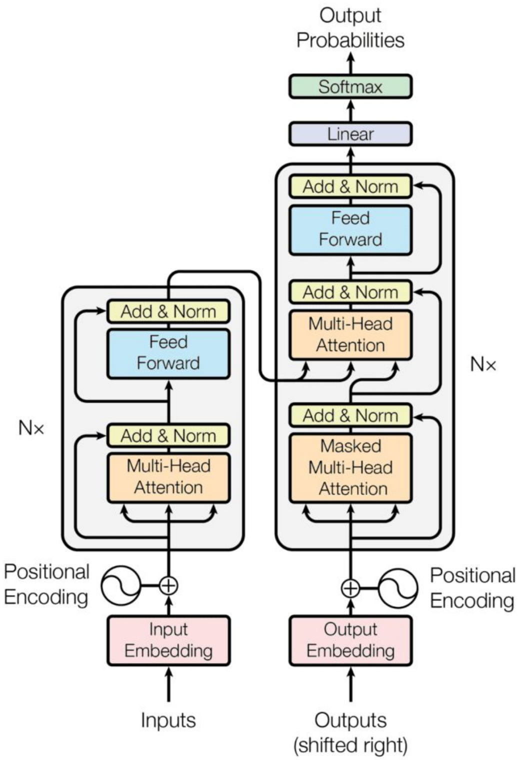


Transformer

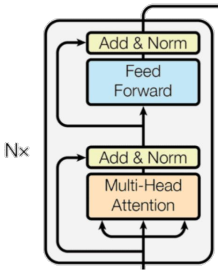


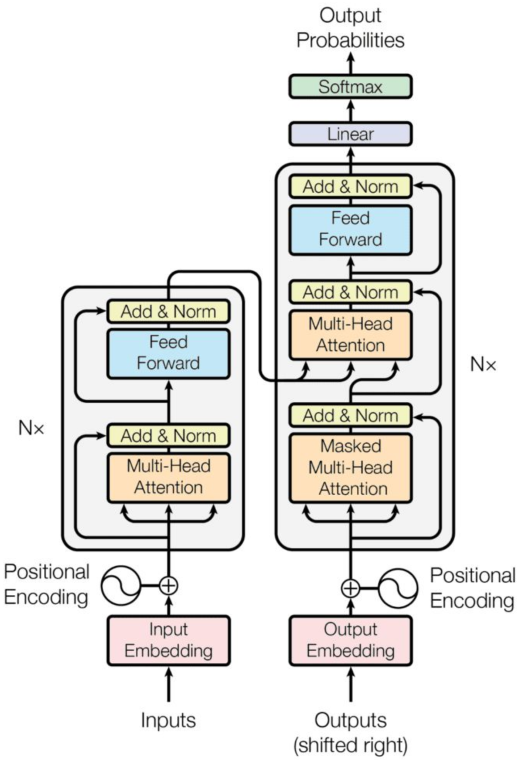


Transformer

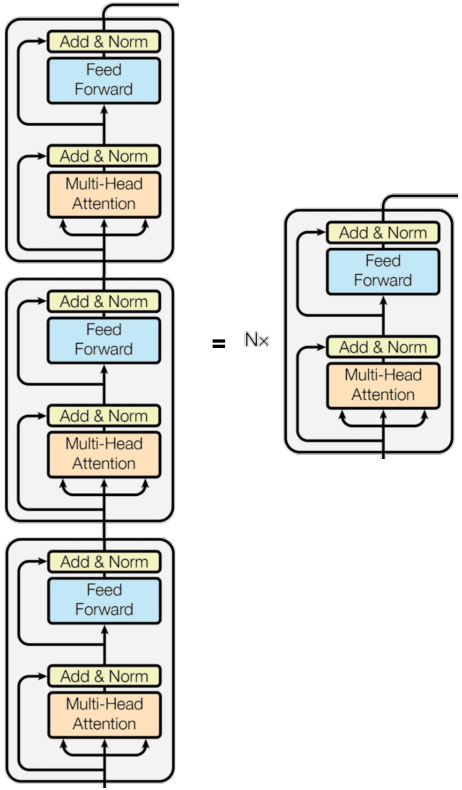


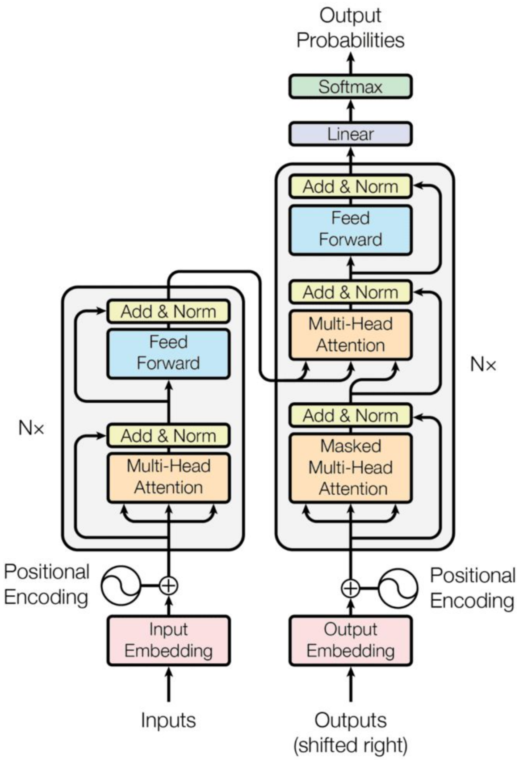
Transformer



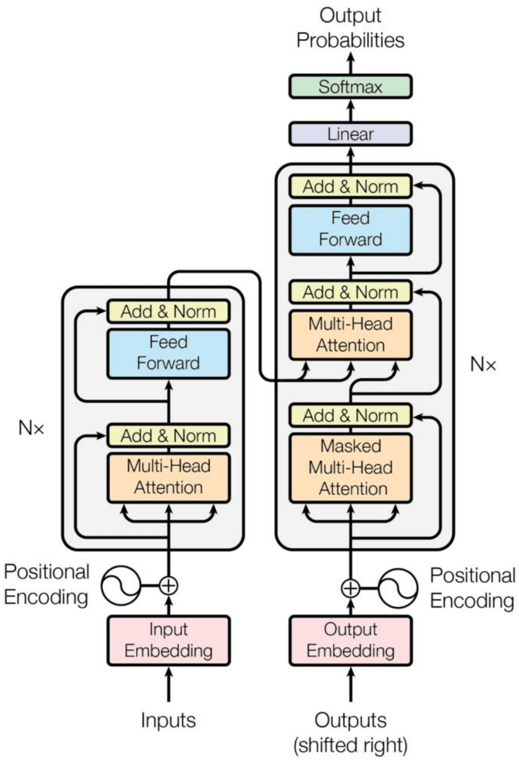


Transformer



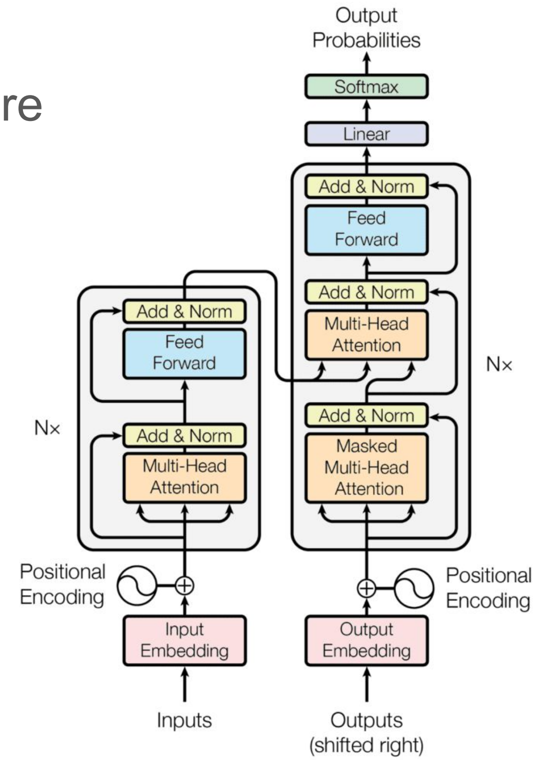


Transformer



Transformer

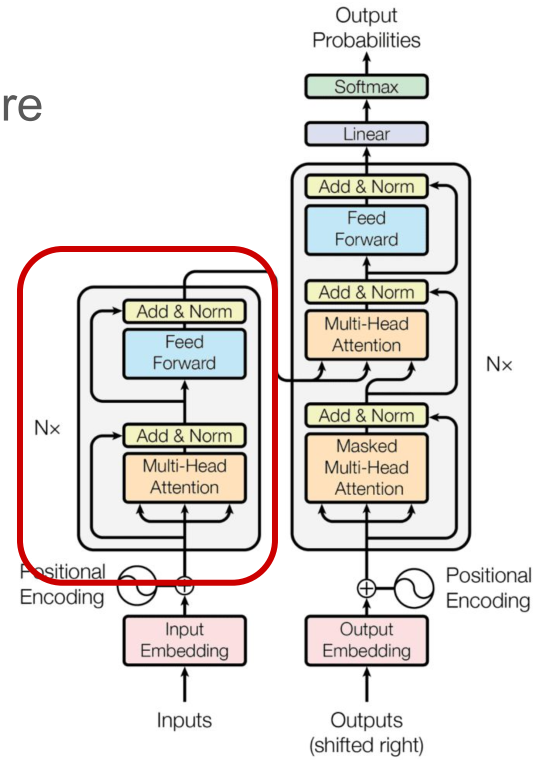
Architecture / Structure



Transformer

Architecture / Structure

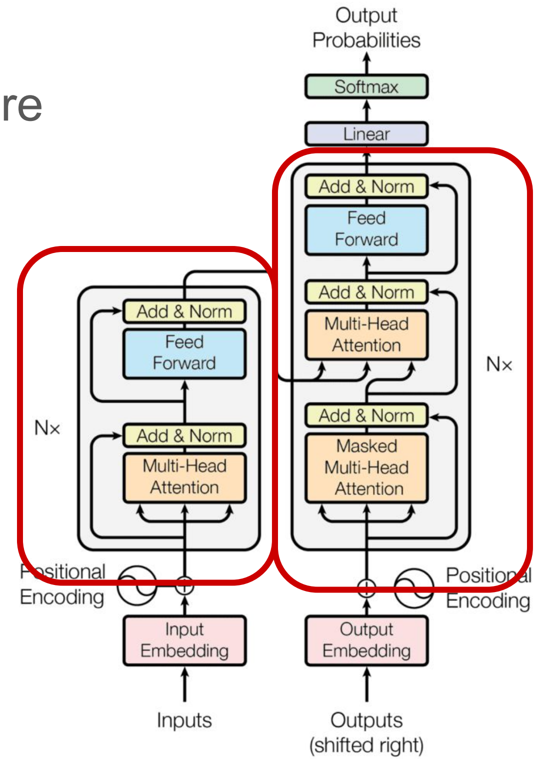
Encoder



Transformer

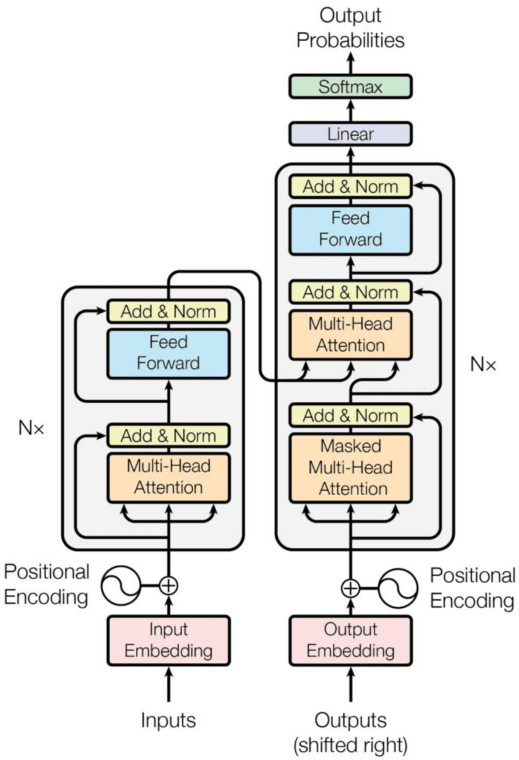
Architecture / Structure

Encoder



Decoder

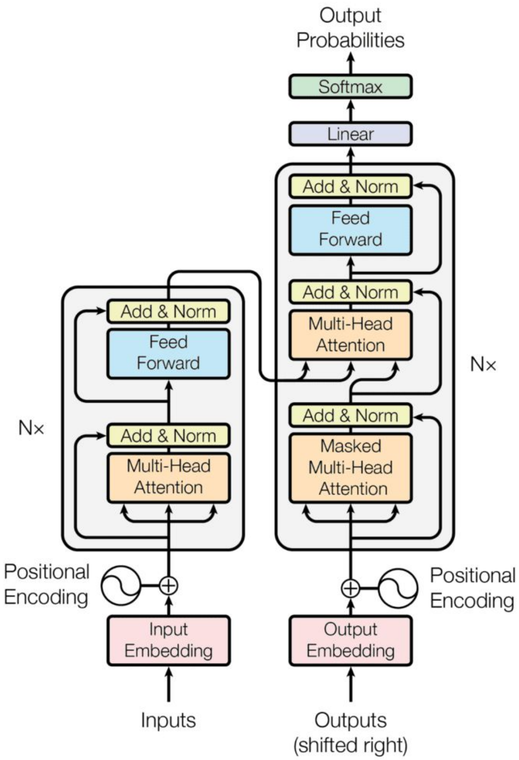
Transformer



Transformer

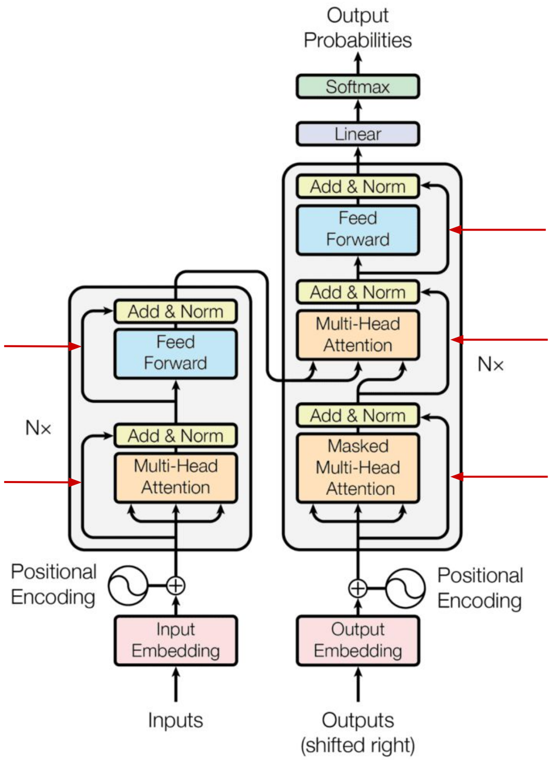
Residual connection:

- skip connection
- shortcut

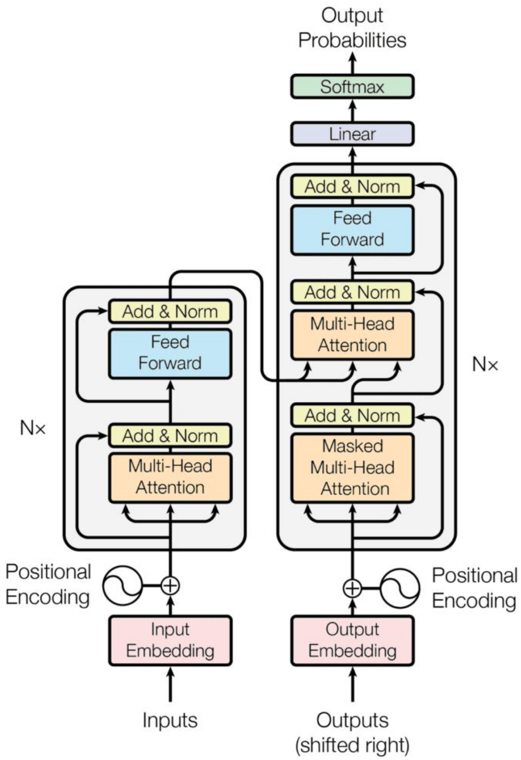


Transformer

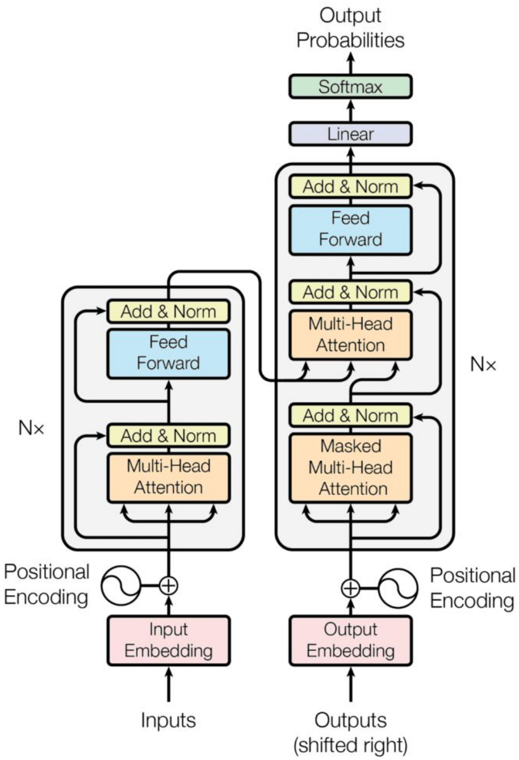
Residual connection:
- skip connection
- shortcut



Transformer



Transformer



Questions?

Transformer: Adv. & Disadv.

Advantages:

- Sequential Processing
    - Attention: O(1)
    - Long-range dependencies
- Parallelization
    - Multi-Head Attention
    - Output sequence:
        - during training
- Flexibility and Generalization
- Scalability
    - Residual connections
- Transfer Learning and Fine-tuning

Disadvantages:

- Resource Intensive:
    - O(N^2) scalability
- Data Hungry
- Complexity in Understanding and Interpretation

Transformer: Adv. & Disadv.

Advantages:

- Sequential Processing
  - Attention: O(1)
  - Long-range dependencies
- Parallelization
  - Multi-Head Attention
  - Output sequence:
    - during training
- Flexibility and Generalization
- Scalability
  - Residual connections
- Transfer Learning and Fine-tuning

Disadvantages:

- Resource Intensive:
  - O(N^2) scalability
- Data Hungry
- Complexity in Understanding and Interpretation

Transformer: Adv. & Disadv.

Advantages:

- Sequential Processing
    - Attention: O(1)
    - Long-range dependencies
- Parallelization
    - Multi-Head Attention
    - Output sequence:
        - during training
- Flexibility and Generalization
- Scalability
    - Residual connections
- Transfer Learning and Fine-tuning

Disadvantages:

- Resource Intensive:
    - O(N^2) scalability
- Data Hungry
- Complexity in Understanding and Interpretation

Transformer versions: BERT, GPT

BERT:

- Encoder-only Transformer
- used for representation extraction

GPT:

- decoder-only Transformer
- used for text generation

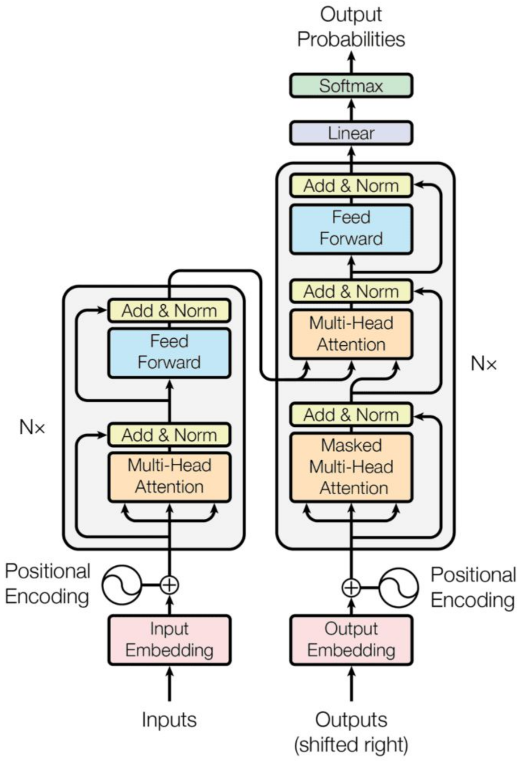
Transformer versions: BERT, GPT

BERT:

- Encoder-only Transformer
- used for representation extraction

GPT:

- decoder-only Transformer
- used for text generation



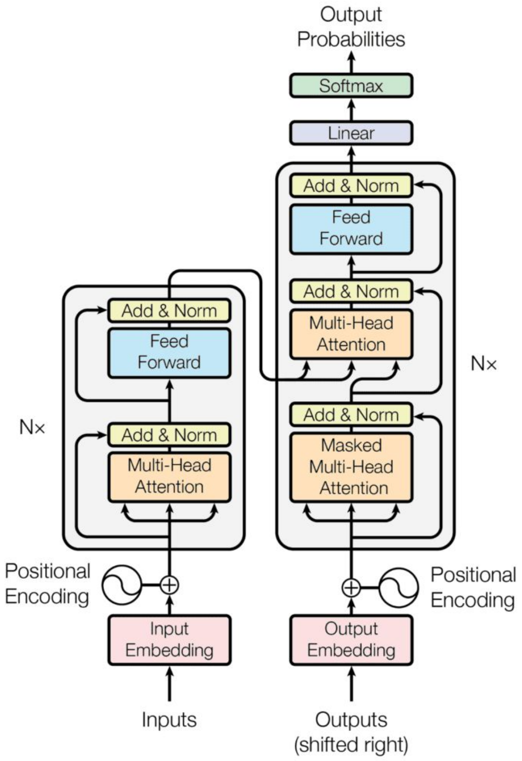
Transformer versions: BERT, GPT

BERT:

- Encoder-only Transformer
- used for representation extraction

GPT:

- decoder-only Transformer
- used for text generation



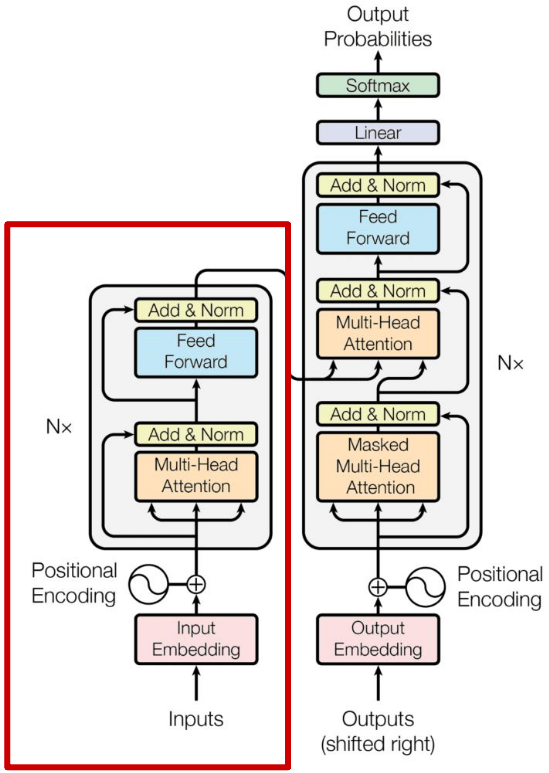
Transformer versions: BERT, GPT

BERT:

- Encoder-only Transformer
- used for representation extraction

GPT:

- decoder-only Transformer
- used for text generation



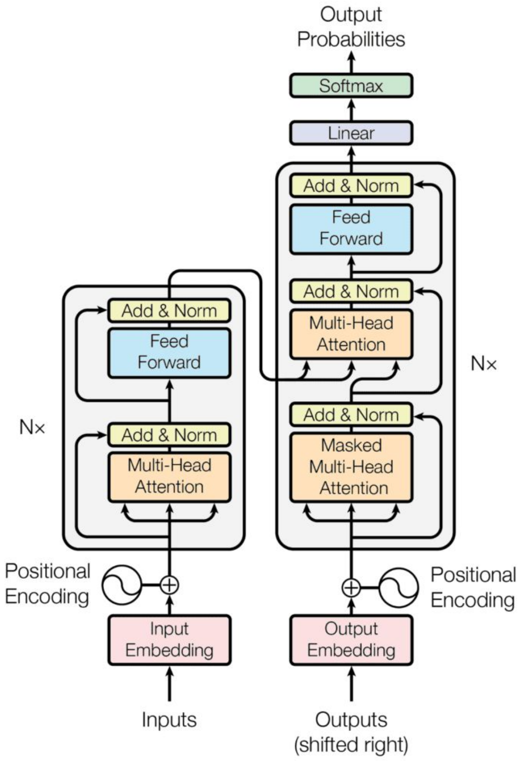
Transformer versions: BERT, GPT

BERT:

- Encoder-only Transformer
- used for representation extraction

GPT:

- decoder-only Transformer
- used for text generation



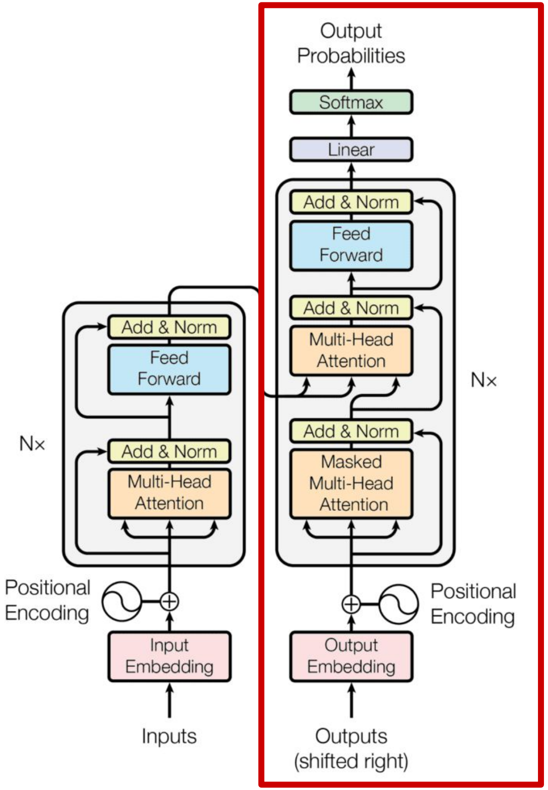
Transformer versions: BERT, GPT

BERT:

- Encoder-only Transformer
- used for representation extraction

GPT:

- decoder-only Transformer
- used for text generation

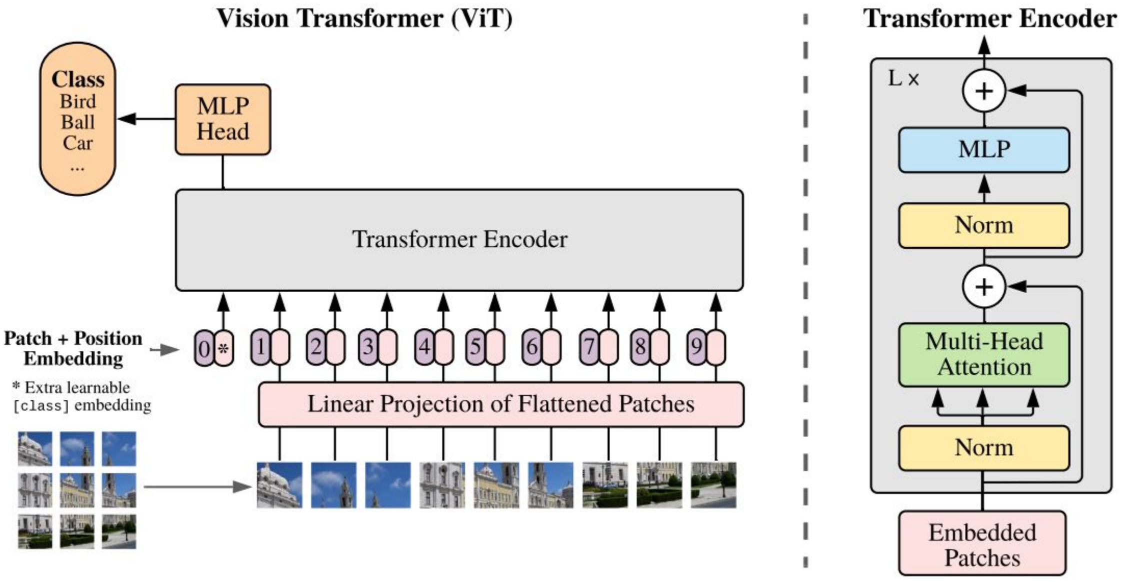


Transformer: Computer Vision

ViT: Vision Transformer

Transformer: Computer Vision

ViT: Vision Transformer



Source

Training “types”

Language Models: Pre-Training and Fine-tuning

Pre-Training:

- Language Modeling objective
- unlabeled dataset from the Internet
- low-quality data

Fine-Tuning:

- downstream task
- specific objective
- high-quality data

Language Models: Pre-Training and Fine-tuning

Pre-Training:

- Language Modeling objective
- unlabeled dataset from the Internet
- low-quality data

Fine-Tuning:

- downstream task
- specific objective
- high-quality data

Language Models: Pre-Training and Fine-tuning

Pre-Training:

- Language Modeling objective
- unlabeled dataset from the Internet
- low-quality data

Fine-Tuning:

- downstream task
- specific objective
- high-quality data

Language Models: Instruction Tuning & Alignment Tuning

Instruction Tuning:

- giving instructions to the Language Model
- improving the Language Model's ability to understand and respond accurately to the user instructions

Alignment Tuning:

- aligning Language Models with human values
  - helpful
  - honest
  - harmless

Language Models: Instruction Tuning & Alignment Tuning

Instruction Tuning:

- giving instructions to the Language Model
- improving the Language Model’s ability to understand and respond accurately to the user instructions

Alignment Tuning:

- aligning Language Models with human values
  - helpful
  - honest
  - harmless

Language Models: Instruction Tuning & Alignment Tuning

Instruction Tuning:

- giving instructions to the Language Model
- improving the Language Model's ability to understand and respond accurately to the user instructions

Alignment Tuning:

- aligning Language Models with human values
  - helpful
  - honest
  - harmless

Evaluation

Evaluation

Considerations:

- not one-to-one mapping
  - one-to-many; many-to-one
  - e.g., Translation, Summarization
- more sophisticated evaluation metrics are required

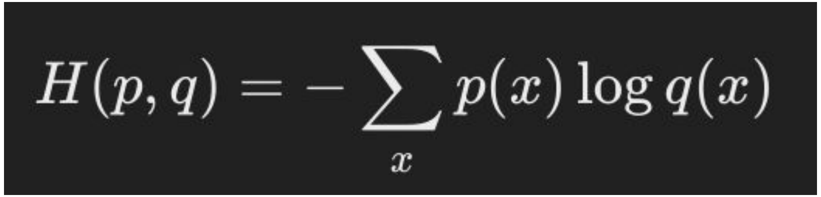
Evaluation: Metrics

Evaluation Metrics:

Evaluation metrics are quantitative tools used to measure the performance of a model.

- Cross-Entropy
- Perplexity
- Edit distance
- CER
- WER
- Accuracy
- F1 Score
- BLEU
- ROUGE
- METEOR
- BERTScore
- CIDEr

Evaluation: Metrics



$$H(p,q) = -\sum_{x} p(x)\log q(x)$$

Evaluation Metrics:

Evaluation metrics are quantitative tools used to measure the performance of a model.

- Cross-Entropy:
- Perplexity
- Edit distance
- CER
- WER
- Accuracy
- F1 Score
- BLEU
- ROUGE
- METEOR
- BERTScore
- CIDEr

Cross-Entropy:

- measuring the difference between the predicted and the true probability distributions

Tasks:

- Classification
- Language Modeling
- Text Generation
- Machine Translation

Evaluation: Metrics

$$PP = 2^{H(p,q)}$$

$$H(p,q) = -\frac{1}{N}\sum_{i=1}^{N}\log_2 q(x_i)$$

Evaluation Metrics:

Evaluation metrics are quantitative tools used to measure the performance of a model.

- Cross-Entropy
- Perplexity:
- Edit distance
- CER
- WER
- Accuracy
- F1 Score
- BLEU
- ROUGE
- METEOR
- BERTScore
- CIDEr

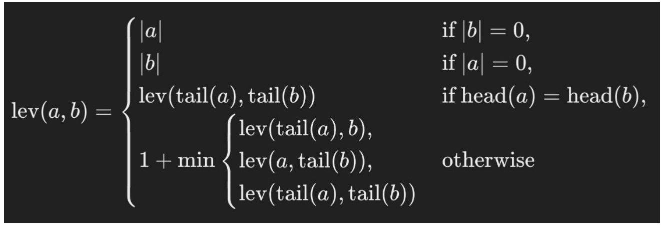
Perplexity:

- evaluating the predictive power of a Language Model
- the exponentiation of the average log-likelihood of a sequence
- the lower the better
- how well a probability model predicts a sample

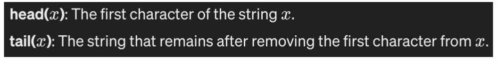
Tasks:

- Language Modeling
- Text Generation

Evaluation: Metrics



$$\operatorname{lev}(a,b) = \begin{cases} |a| & \text{if } |b| = 0, \\ |b| & \text{if } |a| = 0, \\ \operatorname{lev}(\operatorname{tail}(a), \operatorname{tail}(b)) & \text{if } \operatorname{head}(a) = \operatorname{head}(b), \\ 1 + \min \begin{cases} \operatorname{lev}(\operatorname{tail}(a), b), \\ \operatorname{lev}(a, \operatorname{tail}(b)), \\ \operatorname{lev}(\operatorname{tail}(a), \operatorname{tail}(b)) \end{cases} & \text{otherwise} \end{cases}$$



head(x): The first character of the string x.

tail(x): The string that remains after removing the first character from x.

Evaluation Metrics:

Evaluation metrics are quantitative tools used to measure the performance of a model.

- Cross-Entropy
- Perplexity
- Edit distance:
- CER
- WER
- Accuracy
- F1 Score
- BLEU
- ROUGE
- METEOR
- BERTScore
- CIDEr

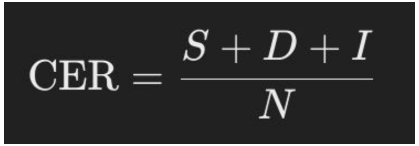
Edit Distance aka. Levenshtein Distance:

- quantifying the dissimilarity between 2 strings
- counting the minimum number of operations (insertion, deletion, substitution) required to transform one string into the other

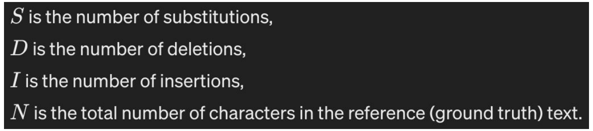
Tasks:

- Optical Character Recognition (OCR)

Evaluation: Metrics



$$\mathrm{CER} = \frac{S + D + I}{N}$$



S is the number of substitutions,

D is the number of deletions,

I is the number of insertions,

N is the total number of characters in the reference (ground truth) text.

Evaluation Metrics:

Evaluation metrics are quantitative tools used to measure the performance of a model.

- Cross-Entropy
- Perplexity
- Edit distance
- CER:
- WER
- Accuracy
- F1 Score
- BLEU
- ROUGE
- METEOR
- BERTScore
- CIDEr

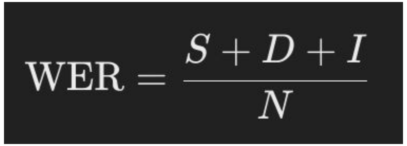
CER: Character Error Rate:

- measuring the number of character-level errors (insertion, deletion, substitution)
- between text sequences

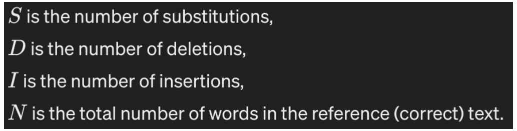
Tasks:

- Optical Character Recognition (OCR)

Evaluation: Metrics



$$\mathrm{WER} = \frac{S+D+I}{N}$$



S is the number of substitutions,

D is the number of deletions,

I is the number of insertions,

N is the total number of words in the reference (correct) text.

Evaluation Metrics:

Evaluation metrics are quantitative tools used to measure the performance of a model.

- Cross-Entropy
- Perplexity
- Edit distance
- CER
- WER:
- Accuracy
- F1 Score
- BLEU
- ROUGE
- METEOR
- BERTScore
- CIDEr

WER: Word Error Rate

- measuring the number of word-level errors
- between text sequences

Tasks:

- Optical Character Recognition (OCR)

Evaluation: Metrics

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Evaluation Metrics:

Evaluation metrics are quantitative tools used to measure the performance of a model.

- Cross-Entropy
- Perplexity
- Edit distance
- CER
- WER
- Accuracy:
- F1 Score
- BLEU
- ROUGE
- METEOR
- BERTScore
- CIDEr

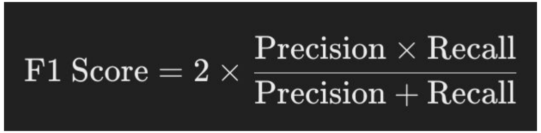
Accuracy:

- measuring the proportion of correct predictions out of the total predictions

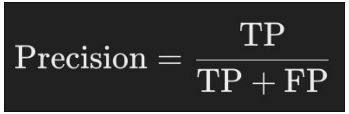
Tasks:

- Text Classification
- Sentiment Analysis
- Named Entity Recognition
- Information Retrieval

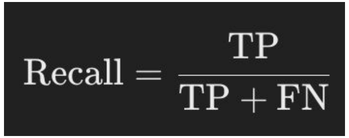
Evaluation: Metrics



$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$



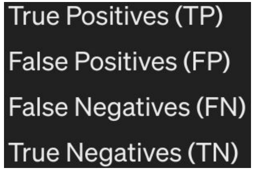
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$



$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Evaluation Metrics:

Evaluation metrics are quantitative tools used to measure the performance of a model.



True Positives (TP)

False Positives (FP)

False Negatives (FN)

True Negatives (TN)

- Cross-Entropy
- Perplexity
- Edit distance
- CER
- WER
- Accuracy
- F1 Score:
- BLEU
- ROUGE
- METEOR
- BERTScore
- CIDEr

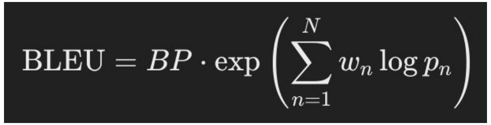
F1 Score:

- a harmonic mean of Precision and Recall
- used in classification to balance the trade-off between false positives and false negatives
- Precision:
  - measuring the ratio of correctly predicted positive observations to the total predicted positives
- Recall:
  - measuring the ratio of correctly predicted positive observations to all actual positives

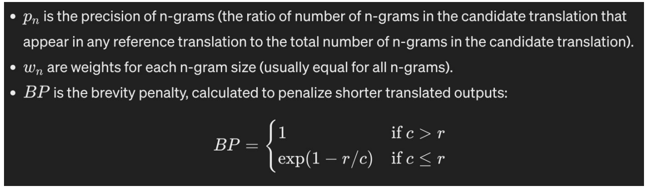
Tasks:

- Information Retrieval

Evaluation: Metrics



$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$



- p_n is the precision of n-grams (the ratio of number of n-grams in the candidate translation that appear in any reference translation to the total number of n-grams in the candidate translation).
- w_n are weights for each n-gram size (usually equal for all n-grams).
- BP is the brevity penalty, calculated to penalize shorter translated outputs:

$$BP = \begin{cases} 1 & \text{if } c > r \\ \exp(1 - r/c) & \text{if } c \leq r \end{cases}$$

Evaluation Metrics:

Evaluation metrics are quantitative tools used to measure the performance of a model.

- Cross-Entropy
- Perplexity
- Edit distance
- CER
- WER
- Accuracy
- F1 Score
- BLEU:
- ROUGE
- METEOR
- BERTScore
- CIDEr

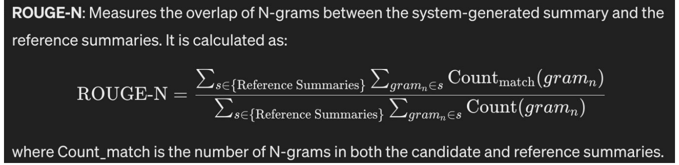
BLEU: Bilingual Evaluation Understudy

- a precision-oriented metric for evaluating Machine Translation quality by comparing n-gram overlaps between the machine-generated and reference translations

Tasks:

- Machine Translation
- Text Summarization (less common)

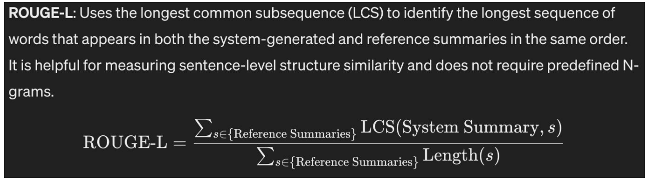
Evaluation: Metrics



ROUGE-N: Measures the overlap of N-grams between the system-generated summary and the reference summaries. It is calculated as:

$$\text{ROUGE-N} = \frac{\sum_{s\in\{\text{Reference Summaries}\}} \sum_{gram_n \in s} \text{Count}_{\text{match}}(gram_n)}{\sum_{s\in\{\text{Reference Summaries}\}} \sum_{gram_n \in s} \text{Count}(gram_n)}$$

where Count_match is the number of N-grams in both the candidate and reference summaries.



ROUGE-L: Uses the longest common subsequence (LCS) to identify the longest sequence of words that appears in both the system-generated and reference summaries in the same order. It is helpful for measuring sentence-level structure similarity and does not require predefined N-grams.

$$\text{ROUGE-L} = \frac{\sum_{s\in\{\text{Reference Summaries}\}} \text{LCS}(\text{System Summary}, s)}{\sum_{s\in\{\text{Reference Summaries}\}} \text{Length}(s)}$$

Evaluation Metrics:

Evaluation metrics are quantitative tools used to measure the performance of a model.

- Cross-Entropy
- Perplexity
- Edit distance
- CER
- WER
- Accuracy
- F1 Score
- BLEU
- ROUGE:
- METEOR
- BERTScore
- CIDEr

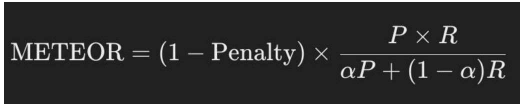
ROUGE: Recall-Oriented Understudy for Gisting Evaluation

- a recall-focused metric used to evaluate the quality of summaries by measuring the overlap of n-grams, word sequences, and word pairs between the automated summary and a set of reference summaries

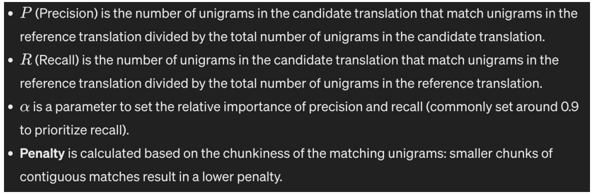
Tasks:

- Text Summarization
- Machine Translation (less common)

Evaluation: Metrics



$$\text{METEOR} = (1 - \text{Penalty}) \times \frac{P \times R}{\alpha P + (1 - \alpha)R}$$



- P (Precision) is the number of unigrams in the candidate translation that match unigrams in the reference translation divided by the total number of unigrams in the candidate translation.
- R (Recall) is the number of unigrams in the candidate translation that match unigrams in the reference translation divided by the total number of unigrams in the reference translation.
- α is a parameter to set the relative importance of precision and recall (commonly set around 0.9 to prioritize recall).
- **Penalty** is calculated based on the chunkiness of the matching unigrams: smaller chunks of contiguous matches result in a lower penalty.

Evaluation Metrics:

Evaluation metrics are quantitative tools used to measure the performance of a model.

- Cross-Entropy
- Perplexity
- Edit distance
- CER
- WER
- Accuracy
- F1 Score
- BLEU
- ROUGE
- METEOR:
- BERTScore
- CIDEr

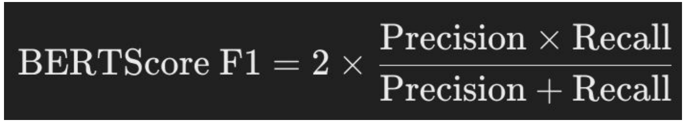
METEOR:

- combining precision, recall, and synonym matching to evaluate translation quality
- also considering morphological variations and paraphrasing to achieve a more nuanced assessment

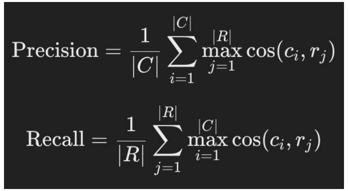
Tasks:

- Machine Translation
- Paraphrase detection

Evaluation: Metrics

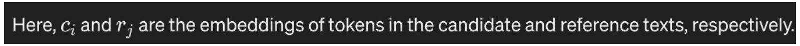


$$\text{BERTScore F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$



$$\text{Precision} = \frac{1}{|C|} \sum_{i=1}^{|C|} \max_{j=1}^{|R|} \cos(c_i, r_j)$$

$$\text{Recall} = \frac{1}{|R|} \sum_{j=1}^{|R|} \max_{i=1}^{|C|} \cos(c_i, r_j)$$



Here, c_i and r_j are the embeddings of tokens in the candidate and reference texts, respectively.

Evaluation Metrics:

Evaluation metrics are quantitative tools used to measure the performance of a model.

- Cross-Entropy
- Perplexity
- Edit distance
- CER
- WER
- Accuracy
- F1 Score
- BLEU
- ROUGE
- METEOR
- BERTScore:
- CIDEr

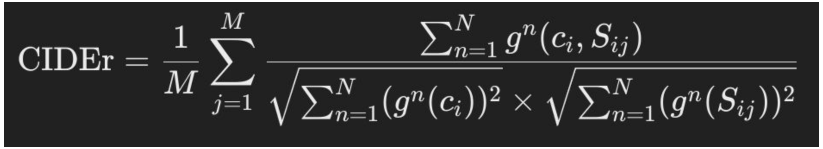
BERTScore:

- utilizing BERT contextual embeddings to match words and phrases between predicted and reference texts
- providing a token-level similarity score

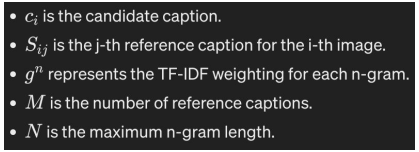
Tasks:

- Machine Translation
- Text Generation
- Text Summarization

Evaluation: Metrics



$$\text{CIDEr} = \frac{1}{M}\sum_{j=1}^{M} \frac{\sum_{n=1}^{N} g^n(c_i, S_{ij})}{\sqrt{\sum_{n=1}^{N}(g^n(c_i))^2} \times \sqrt{\sum_{n=1}^{N}(g^n(S_{ij}))^2}}$$



- c_i is the candidate caption.
- S_{ij} is the j-th reference caption for the i-th image.
- g^n represents the TF-IDF weighting for each n-gram.
- M is the number of reference captions.
- N is the maximum n-gram length.

Evaluation Metrics:

Evaluation metrics are quantitative tools used to measure the performance of a model.

- Cross-Entropy
- Perplexity
- Edit distance
- CER
- WER
- Accuracy
- F1 Score:
- BLEU
- ROUGE
- METEOR
- BERTScore
- CIDEr:

CIDEr: Consensus-based Image Description Evaluation

- designed for evaluating the quality of image descriptions
- emphasizing the importance of capturing salient details
  - likely mentioned by humans when describing the image
- calculating similarity scores based on the TF-IDF weighting for each n-gram in the candidate caption relative to a reference corpus of captions

Tasks:

- Image Captioning

Evaluation: Benchmarks (datasets)

Benchmarks:

Benchmarks are standard points of reference against which the performance of a model can be compared or assessed.

Benchmarks often consists of predefined datasets, tasks, or a set of performance metrics that have been widely accepted by a community as a basis for comparison.

- **GLUE**
- **SuperGLUE**
- **SQuAD**
- **RACE**

Evaluation: Benchmarks (datasets)

Benchmarks:

Benchmarks are standard points of reference against which the performance of a model can be compared or assessed.

Benchmarks often consists of predefined datasets, tasks, or a set of performance metrics that have been widely accepted by a community as a basis for comparison.

- **GLUE:**
- **SuperGLUE**
- **SQuAD**
- **RACE**

GLUE: General Language Understanding Evaluation

- 9 sentence- or sentence-pair language understanding tasks
- providing a single-number metric

Evaluation: Benchmarks (datasets)

Source

GLUE: Datasets

- **CoLA**: Acceptability (The Corpus of Linguistic Acceptability)
  - They made him angry. (1 = acceptable)
  - This building is than that one. (0 = unacceptable)
- **SST-2**: Sentiment (Stanford Sentiment Treebank)
  - The movie as a whole is cheap junk and an insult to their death-defying efforts (= 0.111)
  - The movie is funny, smart, visually inventive, and most of all, alive. (=0.930)
  - Has both. (=0.5)
- **MRPC**: Paraphrase (The Microsoft Research Paraphrase Corpus)
  - Yesterday, Taiwan reported 35 new infections, bringing the total number of cases to 418.
  - The island reported another 35 probable cases yesterday, taking its total to 418.
- **STS-B**: Sentence Similarity (Semantic Textual Similarity Benchmark)
  - Elephants are walking down a trail. A herd of elephants are walking along a trail. (4.6)
  - A man is making a bed. A woman is playing a guitar. (0.0)
- **QQP**: Paraphrase (The Quora Question Pairs3)
  - How can I be a good geologist? What should I do to be a great geologist? (1)
  - How can I increase the speed of my internet connection while using a VPN? How can Internet speed be increased by hacking through DNS? (0)
- **MNLI-m**: Natural Language Inference
  - Jon walked back to the town to the smithy. Jon traveled back to his hometown. (1 = neutral)
  - Tourist Information offices can be very helpful. Tourist Information offices are never of any help. (2 = contradiction)
  - I’m confused. Not all of it is very clear to me. (0 = entailed)
- **QNLI**: Question Answering / Natural Language Inference (The Stanford Question Answering Dataset - SQuAD)
  - How was the Everton FC’s crest redesign received by fans? The redesign was poorly received by supporters, with a poll on an Everton fan site registering a 91% negative response to the crest. (0 = answerable)
  - In what year did Robert Louis Stevenson die? Mission work in Samoa had begun in the late 1830 by John Williams. (1 = not answerable)
- **RTE**: Natural Language Inference (The Recognizing Textual Enttailment)
  - The marriage is planned to take place in the Kiev Monastery of the Caves, whose father superior, Bishop Pavel, is Yulia Timoshenko’s spiritual father. Yulia Timoshenko is the daughter of Bishop Pavel. (1 = not entailed)
  - With its headquarters in Madrid, Spain, WTO is an inter-governmental body entrusted by the United nations to promote and develop tourism. The WTO headquarters is located in Madrid, Spain. (0 = entailed)
- **WNLI**: Coreference / Natural Language Inference (The Winograd Schema Challenge)
  - The trophy didn’t fit into the suitcase because it was too [large/small]. Question: What was too [large/small]? Answer: the trophy / the suitcase.
  - Lily spoke to Donna, breaking her concentration. Lily spoke to Donna, breaking Lily’s concentration. (0)
  - I put the cake away in the refrigerator. It has a lot of leftovers in it. The refrigerator has a lot of leftovers in it. (1)

Evaluation: Benchmarks (datasets)

Benchmarks:

Benchmarks are standard points of reference against which the performance of a model can be compared or assessed.

Benchmarks often consists of predefined datasets, tasks, or a set of performance metrics that have been widely accepted by a community as a basis for comparison.

- **GLUE**
- **SuperGLUE:**
- **SQuAD**
- **RACE**

SuperGLUE:

- a new set of more difficult language understanding tasks

Evaluation: Benchmarks (datasets)

Benchmarks:

Benchmarks are standard points of reference against which the performance of a model can be compared or assessed.

Benchmarks often consists of predefined datasets, tasks, or a set of performance metrics that have been widely accepted by a community as a basis for comparison.

- **GLUE**
- **SuperGLUE**
- **SQuAD**:
- **RACE**

SQuAD: Stanford Question Answering Dataset

- a reading comprehension dataset
- consisting of questions on a set of Wikipedia articles

Evaluation: Benchmarks (datasets)

Benchmarks:

Benchmarks are standard points of reference against which the performance of a model can be compared or assessed.

Benchmarks often consists of predefined datasets, tasks, or a set of performance metrics that have been widely accepted by a community as a basis for comparison.

- **GLUE**
- **SuperGLUE**
- **SQuAD**
- **RACE:**

RACE: ReAding Comprehension datasets from Examinations

- collected from English exams
- covering a variety of topics

Frameworks

Frameworks

Frameworks:

- PyTorch
- TensorFlow
- JAX

Libraries:

- Hugging Face
- NLTK (Natural Language Toolkit)
- spaCy
- HuSpaCy
- Fairseq
- NielsRogge

Model Hubs:

- Hugging Face

Additional Resources

Transformer:

- https://www.tensorflow.org/text/tutorials/transformer

Attention:

-