بسم الله الرحمن الرحيم

Transformer Language Model

Ehsaneddin Asgari

easgari [at] hbku [dot] edu [dot] qa

Mena-ML Winterschool
Doha, February 2025

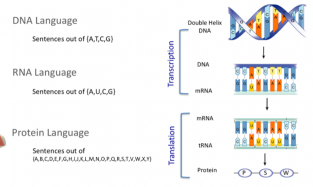
QCRI

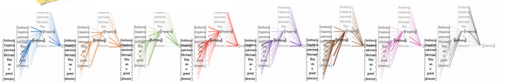
Table of Contents

What is Language?

1 Why Language Modeling?

"A language is a collection of sentences of finite length all constructed from a finite alphabet of symbols.."
(Chomsky, 1959)





Distributional Hypothesis

Definition: Words that occur in similar contexts tend to have similar meanings **(Firth, 1950)**

$$\text{sim}(w_1, w_2) \approx \text{sim}(\text{context}(w_1), \text{context}(w_2))$$

Basis for Word Emb., LMs, LLMs.

What is Language Modeling?

1 Why Language Modeling?

Definition: A language model (LM) assigns probabilities to sequences of words.

$$P(w_1, w_2, ..., w_T) = \prod_{t=1}^{T} P(w_t | w_1, ..., w_{t-1})$$

Language Model:

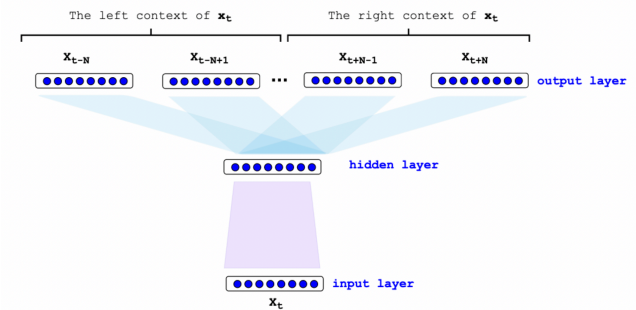
- Syntax and semantics of language
- Compressed version of the language

Types of LMs:

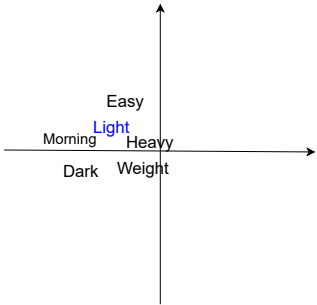
- **N-gram models:** Markov assumption, estimating the conditional prob.
- **Neural LMs:** Learn representations from data.

Context-indep. Word Representation

1 Why Language Modeling?



Light



Projection of embedding space into 2D

- The morning **light** filled the land.
- This suitcase is very **light** and easy to carry.
- He made a **light** joke to ease the tension.

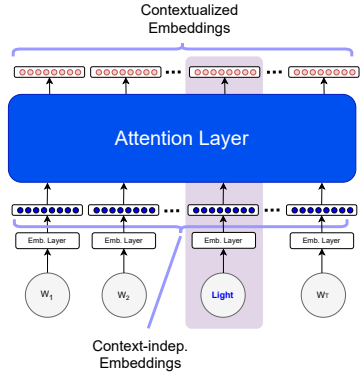
Table of Contents

Multiple Meanings of “Light” in Transformers?

2 Transformer Language Model

Sense Example	
	The morning **light** filled the land..
	This suitcase is very **light** and easy to carry.
	He made a **light** joke to ease the tension.

Attention Concept

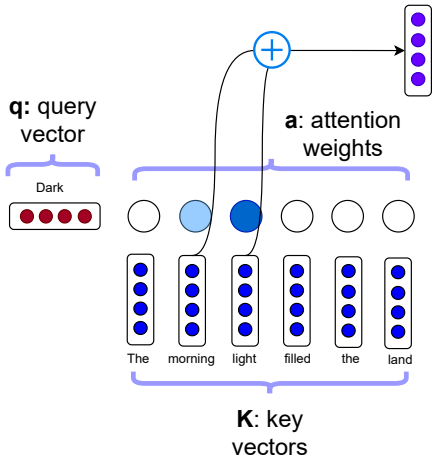


Cross-Attention

$$\mathbf{a} = \text{softmax}\left(\frac{\mathbf{q}K^T}{\sqrt{d_k}}\right)$$

$$A(\mathbf{q}, K, V) = \mathbf{a}V$$

- Attention vector $\mathbf{a}$ assigns weights to keys.
- “light”: high attention due to contrast to “dark”.
- The weighted sum of “keys” determines the output.

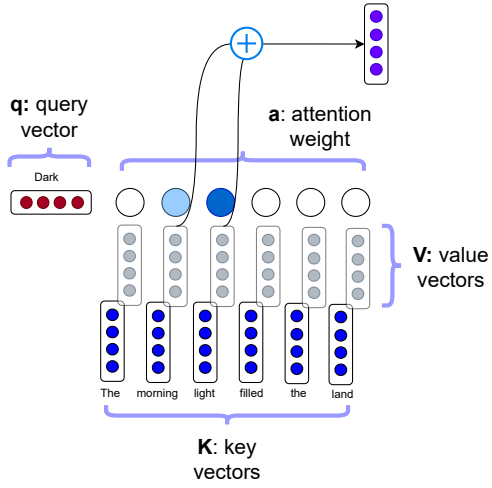


Cross-Attention

$$\mathbf{a} = \text{softmax}\left(\frac{\mathbf{q}K^T}{\sqrt{d_k}}\right)$$

$$A(\mathbf{q}, K, V) = \mathbf{a}V$$

- Attention vector $\mathbf{a}$ assigns weights to keys.
- "light": high attention due to contrast to "dark".
- The weighted sum of "keys" or "values" determines the output.
- What are potential applications?



Some Key Questions on Attention

1. How does the output relate to q and V?
2. Why use Softmax instead of direct normalization?

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

3. Why is $\sqrt{d_k}$ included in **softmax** $\left(\frac{\mathbf{q}K^T}{\sqrt{d_k}}\right)$?
4. How can we efficiently compute it for multiple qs at the same time?
5. Can qK^T capture multiple aspects of similarity between the query and keys?

▸ self-attention!

Relation of Output, q, and V

- The output is a weighted sum of values V, with weights derived from attention scores based on the relevance of keys to q.

$$A(\mathbf{q}, K, V) = \text{softmax}\left(\frac{\mathbf{q}K^T}{\sqrt{d_k}}\right)V$$

Back to Questions

Efficient Computation for Multiple Queries

- Instead of computing attention for each query separately, we use **batch matrix multiplication**.
- The queries Q are stacked into a matrix, and attention is computed as:

$$A(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Back to Questions

Why Divide by $\sqrt{d_k}$?

Assume $\mathbf{q}$ and $\mathbf{k}$ are unit vectors of dimension d, with independent components:

$$\mathbb{E}[q_i] = \mathbb{E}[k_i] = 0, \quad \mathrm{Var}[q_i] = \mathrm{Var}[k_i] = 1$$

Then, the expectation and variance of the dot product are:

$$\mathbb{E}[\mathbf{q} \cdot \mathbf{k}] = \sum_{i=1}^{d} \mathbb{E}[q_i]\mathbb{E}[k_i] = 0, \quad \mathrm{Var}[\mathbf{q} \cdot \mathbf{k}] = \sum_{i=1}^{d} 1 = d.$$

Finall, we perform z-score normalization, to calculate the relevance of $\mathbf{k}$ to $\mathbf{q}$:

$$\frac{\mathbf{q}^T\mathbf{k}}{\sqrt{d}}.$$

Back to Questions

Multiple Views on $\text{sim}(q, K^T)$?

- **Multi-Head Attention:** Computes attention in multiple subspaces.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O$$

where each head is:

$$\text{head}_i = \text{softmax}\left(\frac{QW_i^Q(KW_i^K)^T}{\sqrt{d_k}}\right)VW_i^V$$

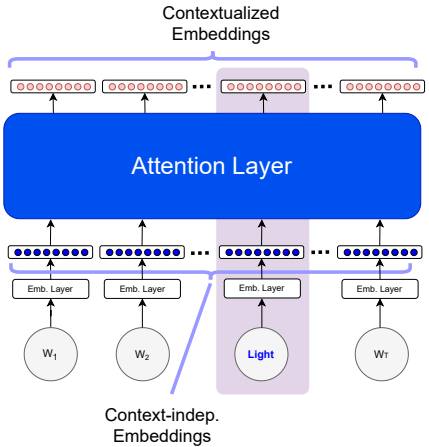
Back to Questions

Self-Attention: Learning Word Relationships

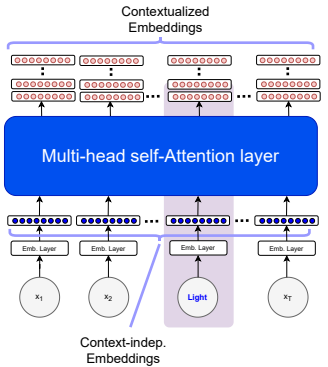
Self-Attention: Computes attention within the same sequence, relating each word to all others.

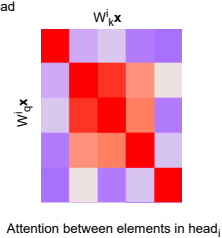
$$A(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- $Q = K[= V]$ (all from input)
- Attention scores: contextual importance.
- Enables long-range dependencies.



Multi-Head Attention





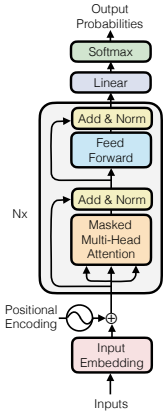
$$\text{head}^i = \text{Attention}(QW_q^i, KW_k^i, VW_v^i)$$

$$A(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}^1,\ldots,\text{head}^h)W^O$$

Transformer Block

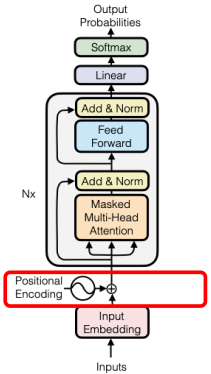
2 Transformer Language Model



Transformer Block:

- Core: multi-head self-attention
- Each layer has layer-norm residual connections.
- Enables parallelization and captures long-range dependencies.
- Stacks multiple layers to improve representation learning.

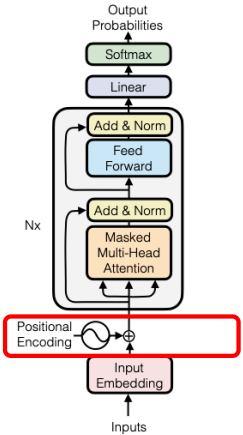
Positional Encoding



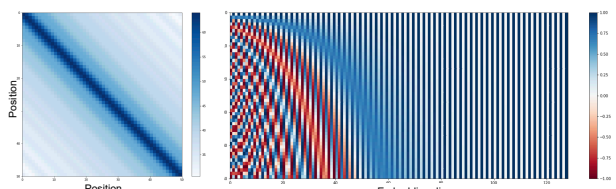
Positional Encoding:

- Transformers lack sequence order.
- Positional embeddings are added to token.
- Uses sinusoidal functions.
- Helps the model differentiate word order.

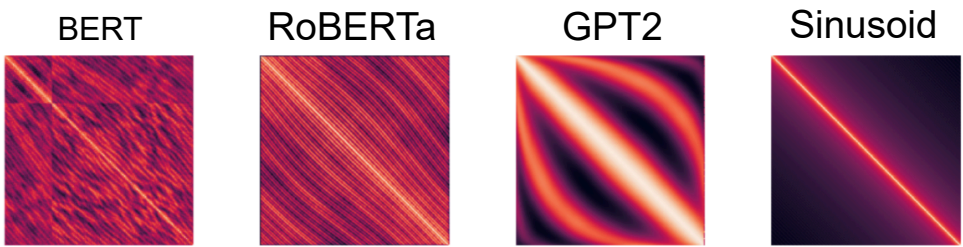
Positional Encoding



$$\vec{p_t}^{(i)} = f(t)^{(i)} := \begin{cases} \sin(\omega_k.t), & \text{if } i = 2k \\ \cos(\omega_k.t), & \text{if } i = 2k+1 \\ \omega_k & = \frac{1}{10000^{2k/d}} \end{cases} \quad (1)$$



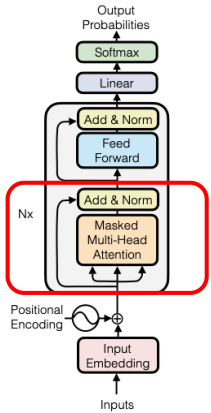
Variation of Positional Encoding



Philipp Dufter, Martin Schmitt, and Hinrich Schütze (2022). “Position Information in Transformers: An Overview.”

Layer Normalization

2 Transformer Language Model



Layer Normalization:

- Normalizes activations across features.
- Stabilizes training.
- Unlike batch norm, it works **independently** for each sample.

Residual Connection

- Prevents vanishing gradients
- Learn better from the input

Layer Normalization

- Batch norm

$a_1, a_2, \ldots, a_B$ (Batch of d-dim vec.)

$$\mu = \frac{1}{B}\sum_{i=1}^{B} a_i, \quad \sigma = \sqrt{\frac{1}{B}\sum_{i=1}^{B}(a_i - \mu)^2}$$

$$\bar{a}_i = \frac{a_i - \mu}{\sigma}$$

- Layer norm

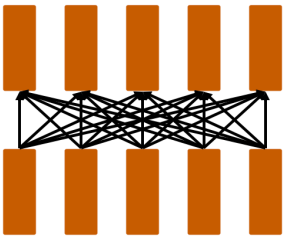
$$\mu = \frac{1}{d}\sum_{i=1}^{d} a_j, \quad \sigma = \sqrt{\frac{1}{d}\sum_{i=1}^{d}(a_j - \mu)^2}$$

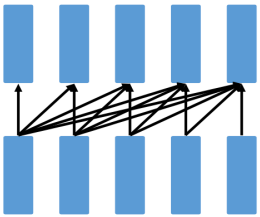
$$\bar{a} = \frac{a - \mu}{\sigma}$$

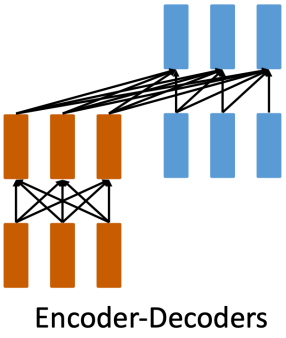
Table of Contents

Transformers Based on Attention Architecture

3 Language Modeling Main Architectures

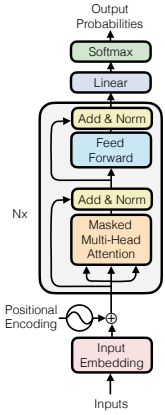






Encoder Model

3 Language Modeling Main Architectures

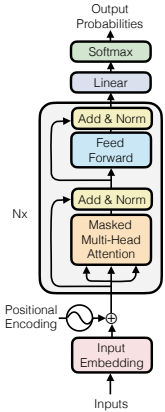


Encoder Model:

- By directional self-attention
- Prediction of masked words at input
- Example: **BERT**, **RoBERTa**

Decoder Model

3 Language Modeling Main Architectures

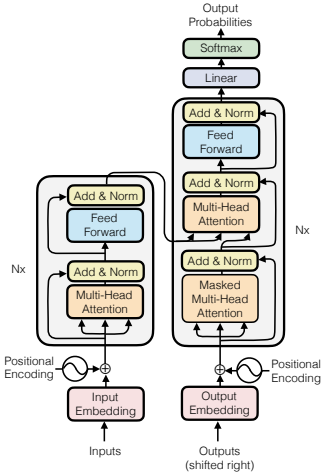


Decoder Model:

- Causal attention (only to previous words)
- Autoregressive: Generates tokens one by one.
- Example: **GPT, LLaMA**

Encoder-Decoder Model

3 Language Modeling Main Architectures

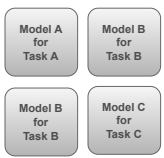


Encoder-Decoder Model:

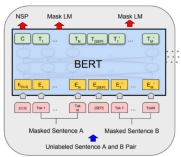
- Uses **cross-attention** to align input and output.
- Encoder **understands input**; Decoder **generates output**
- Example: **T5, BART** (used for translation, summarization).

Paradigm Shift

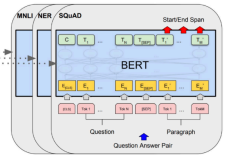
3 Language Modeling Main Architectures



Single Task model



Pre-training



Fine-Tuning

Finetuning

Gemma

Meta

Single model for all capabilities

Table of Contents

Muslim in ML

4 MENA Community

Muslims in Machine Learning:

- Encouraging diversity in AI.
- Community-driven research and collaborations.
- ICLR 2025
- https://www.musiml.org/
- Please reach out: easgari@hbku.edu.qa