Notes on Attention Mechanisms

Abhishek Adhikari

1 Introduction

This document provides an in-depth exploration of the Attention Mechanism, Self-Attention, and Multi-Head Attention, which are foundational to the Transformer architecture introduced in *Attention Is All You Need* (Vaswani et al., 2017). These concepts mark a significant evolution from earlier sequence-to-sequence (seq2seq) models, such as Sutskever et al. (2014), which relied on recurrent neural networks (RNNs) with fixed context vectors. The notes include theoretical foundations, practical examples, and a comparative analysis, designed for researchers, students, and practitioners in natural language processing (NLP).

2 Attention Mechanism

2.1 Definition

The attention mechanism is a technique that enables a model to dynamically focus on specific parts of the input sequence when generating each part of the output sequence. This addresses the limitation of fixed context vectors in traditional seq2seq models, where all input information is compressed into a single vector, often losing detail in longer sequences.

2.2 Mathematical Formulation

The attention mechanism is defined as:

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

- Q (Query): Represents the current token's context or what the model is looking for.
- K (Key): Represents the input tokens' identifiers for matching with the query.
- V (Value): Contains the actual content of the input tokens to be weighted.
- $\sqrt{d_k}$: A scaling factor (where d_k is the dimension of the keys) to prevent large values in high-dimensional spaces, ensuring stable gradient flow during training.

2.3 Process

1. Input embeddings (e.g., word vectors for ”I,” ”am,” ”here”) are transformed into Q, K, and V using learned weight matrices (W_Q, W_K, W_V).
2. The dot product QK^T measures similarity between the query and keys, scaled by $\sqrt{d_k}$.
3. A softmax function converts these scores into attention weights.
4. The weighted sum of values (V) produces a context vector tailored to the current decoding step.

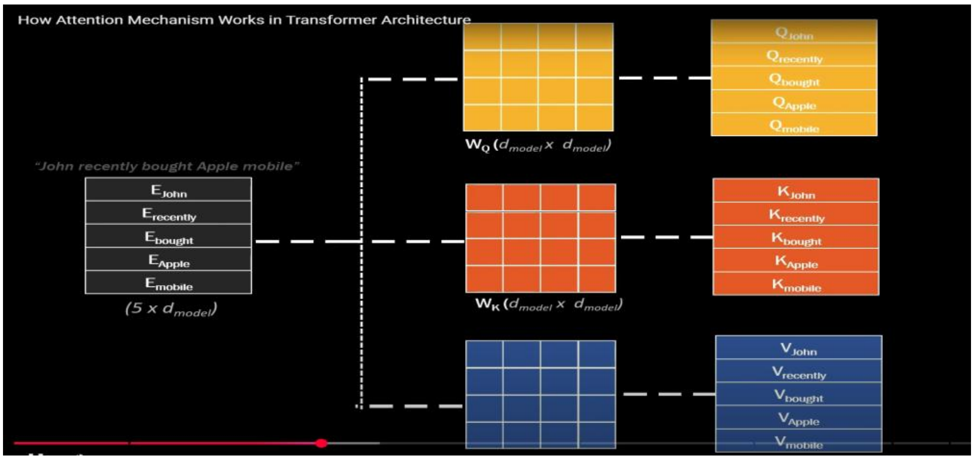
2.4 Example

Consider translating ”I am here” to ”Je suis ici.” The attention mechanism allows the decoder to focus on ”here” when generating ”ici,” rather than relying solely on a fixed context vector as in Sutskever et al. (2014).

2.5 Advantages

- Improves performance over fixed context vectors by preserving more input information.
- Enables better alignment between input and output tokens (e.g., ”here” ↔ ”ici”).
- Lays the groundwork for advanced attention variants in Transformers.

2.6 Visual Aid



3 Self-Attention

3.1 Definition

Self-attention is a specialized form of the attention mechanism where Q, K, and V are derived from the same input sequence. This allows each token to attend to all other tokens within the sequence, capturing intra-sequence relationships.

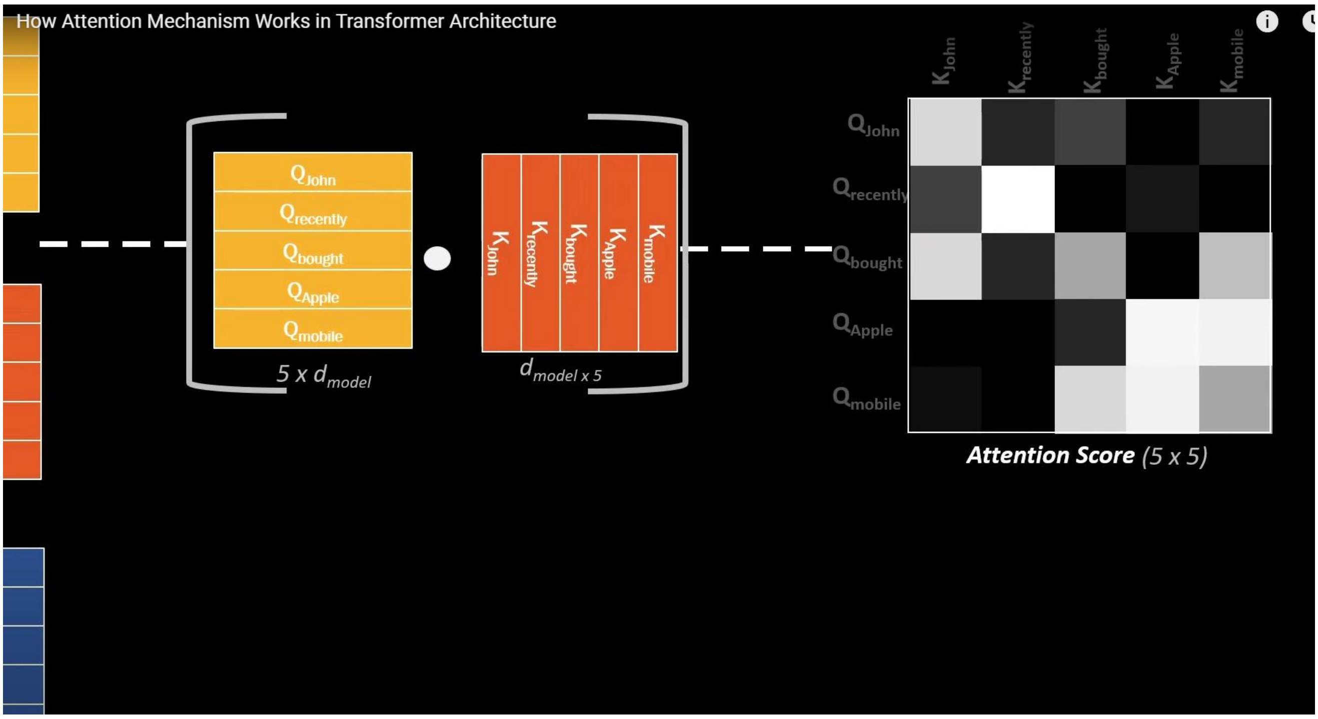
3.2 Process

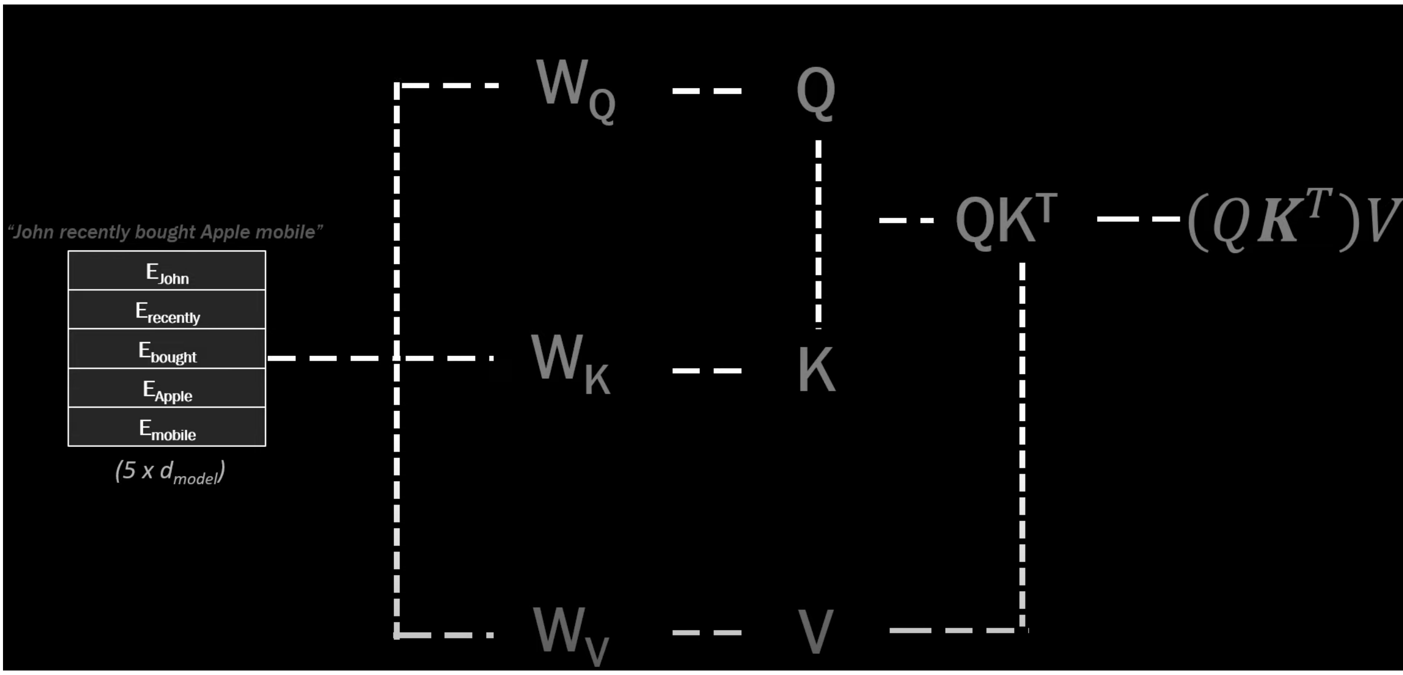
1. Apply weight matrices W_Q, W_K, and W_V to the input embeddings of the entire sequence (e.g., ”John recently bought Apple mobile”).

2. Compute the scaled dot product $\frac{QK^T}{\sqrt{d_k}}$ to generate attention scores.

3. Apply softmax to obtain attention weights, then compute the output as softmax $\left(\frac{QK^T}{\sqrt{d_k}}\right) V$.

4. The result is a set of contextual embeddings reflecting how tokens relate to each other.





3.3 Example

In the sentence ”John bought Apple,” self-attention might reveal that ”bought” attends strongly to ”Apple” (object) and ”John” (subject), capturing syntactic and semantic dependencies.

3.4 Advantages

- Enables parallel processing of all tokens, unlike the sequential nature of LSTMs in Sutskever et al. (2014).
- Captures long-range dependencies (e.g., relating ”John” to ”bought” across the sentence).
- Forms the basis of the Transformer encoder’s ability to model global context.

4 Multi-Head Attention

4.1 Definition

Multi-head attention extends self-attention by running multiple attention operations (heads) in parallel. Each head focuses on different aspects of the input sequence, and their outputs are concatenated and linearly transformed to produce a final representation.

4.2 Process

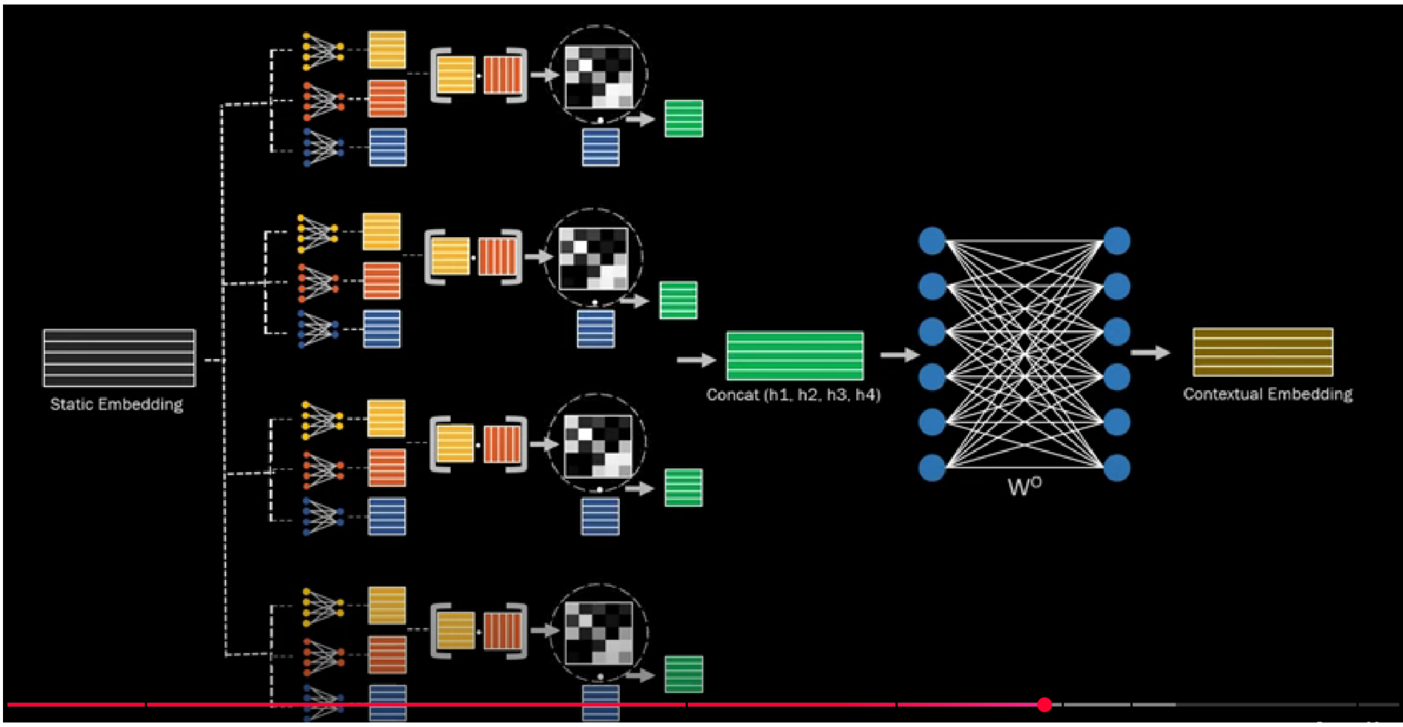
1. Split Q, K, and V into h heads (e.g., 8 heads), each with reduced dimensionality (e.g., $d_k = d_{\text{model}}/h$).
2. For each head i:
   - Compute $Q_i = W_{Q,i}X$, $K_i = W_{K,i}X$, $V_i = W_{V,i}X$.
   - Perform attention: $\text{head}_i = \text{softmax}\left(\frac{Q_iK_i^T}{\sqrt{d_k}}\right)V_i$.
3. Concatenate the heads: $\text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)$.
4. Apply a linear transformation W_O to produce the output.

4.3 Example

For ”I am here” to ”Je suis ici,” one head might focus on syntactic structure (e.g., ”I-am” as subject-verb), while another focuses on semantics (e.g., ”here” as location), improving translation accuracy.

4.4 Advantages

- Captures diverse relationships (e.g., syntax, semantics, long-range dependencies) within a single layer.
- Enhances model expressiveness, leading to higher performance (e.g., BLEU scores of 38-40 vs. 34.81 in Sutskever et al.).
- Replaces sequential LSTM processing with parallel computation, boosting efficiency.



5 Comparative Analysis

5.1 Comparison Table

Aspect	Attention Mechanism	Multi-Head Attention
Definition	Focuses on relevant input parts	Multiple parallel attention heads
Computation	$O(U \cdot T)$ (sequential with RNNs)	$O(n^2 \cdot h)$ (parallel in Transformers)
Example Focus	”here” for ”ici”	Syntax (”I-am”), semantics (”here”)
Performance	Improves Sutskever (36-38 BLEU)	Transformer-level (38-40 BLEU)
Scalability	Limited by RNN sequentiality	Highly scalable with parallel processing
Dependency Modeling	Step-specific context	Global and diverse context

Table 1: Comparison of Attention Mechanism and Multi-Head Attention.

5.2 Key Insights

- The attention mechanism enhances Sutskever et al.’s seq2seq model by introducing dynamic context.

- Self-attention enables Transformers to process sequences in parallel, a major improvement over RNNs.
- Multi-head attention maximizes this potential by integrating multiple perspectives, driving modern NLP advancements.

6 Practical Applications

- **Attention Mechanism**: Improved early machine translation systems (e.g., Google Translate pre-2016).
- **Self-Attention**: Powers Transformer encoders for tasks like text classification and summarization.
- **Multi-Head Attention**: Central to state-of-the-art models (e.g., BERT, GPT) for translation, generation, and more.

7 Conclusion

The evolution from the attention mechanism to self-attention and multi-head attention has transformed NLP, moving from sequential RNN-based models (Sutskever et al., 2014) to parallel, high-performance Transformers. This document, created on October 18, 2025, at 01:04 PM +0545, serves as a comprehensive resource for understanding these advancements.