VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

Logo

Description automatically generated**FACULTY OF INFORMATION TECHNOLOGY**

**NATURAL LANGUAGE PROCESSING FINAL PROJECT**

*Instructor:* **Assoc. Prof. PhD. Lê Anh Cường**

*Student:* **Trương Gia Bảo - 521H0201**

**Vi Thành Đạt - 521H0390**

**Lê Nguyễn Tuyết Nhi – 521H0363**

*Class:* **21H50302**

**21H50301**

**21H50202**

**HO CHI MINH CITY, 2024**

VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

Logo

Description automatically generated**FACULTY OF INFORMATION TECHNOLOGY**

**NATURAL LANGUAGE PROCESSING FINAL PROJECT**

*Instructor:* **Assoc. Prof. PhD. Lê Anh Cường**

*Student:* **Trương Gia Bảo - 521H0201**

**Lê Nguyễn Tuyết Nhi – 521H0363**

*Class:* **21H50302**

**21H50301**

**21H50202**

**HO CHI MINH CITY, 2024**

# ACKNOWLEDGEMENT

I would like to express my sincere thanks to the teachers and lecturers of Ton Duc Thang University and especially Mr. Le Anh Cuong - lecturer of Natural Language Processing Group 03 for kindly helping and supporting as well as answering our questions throughout the process of making this report, so that I can get the best results.

**THE PROJECT WAS COMPLETED**

**AT TON DUC THANG UNIVERSITY**

I would like to assure you that this is my own project and guided by Le Anh Cuong. The research contents and results in this topic are honest and have not been published in any form before. The data in the tables for analysis, comments, and evaluations collected by the author himself from different sources are clearly stated in the references section.

In addition, the project also uses some comments, reviews as well as figures of other authors and other organizations with quotes and annotations of origin.

If any fraud is detected, I would like to take full responsibility for the content of my project. Ton Duc Thang University is not involved in copyright or copyright violations caused by me in the process (if any).

*Ho Chi Minh City, 28th Oct 2024*

*Author*

*Trương Gia Bảo*

*Lê Nguyễn Tuyết Nhi*

*Vi Thành Đạt*

**FACULTY ENDORSEMENTS AND REVIEWS**

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(ký và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(ký và ghi họ tên)

# TABLE OF CONTENTS

[ACKNOWLEDGEMENT 1](#_Toc181131909)

[TABLE OF CONTENTS 1](#_Toc181131910)

[CHAPTER 1: CLASSIFICATION MODEL 2](#_Toc181131911)

[CHAPTER 2: LSTM ( LONG SHORT TERM MEMORY ) 7](#_Toc181131912)

[2.1 LSTM ( Long Short Term Memory ) 7](#_Toc181131913)

[2.2 BiLSTM ( Bidirectional Long Short Term Memory ) 11](#_Toc181131914)

[2.2.1 Bidirectional Processing 12](#_Toc181131915)

[2.2.2 Forward Pass 13](#_Toc181131916)

[2.2.3 Backward Pass 13](#_Toc181131917)

[CHAPTER 3: WORD EMBEDDING 14](#_Toc181131918)

[CHAPTER 4: WORD2VEC 17](#_Toc181131919)

[4.1 Model Architecture 18](#_Toc181131920)

[4.2 Neural Network Training 19](#_Toc181131921)

[4.3 Vector Representations 20](#_Toc181131922)

[4.4 Advantage and Disadvantage 20](#_Toc181131923)

[REFERENCE 21](#_Toc181131924)

# CHAPTER 1: TRANSFORMER

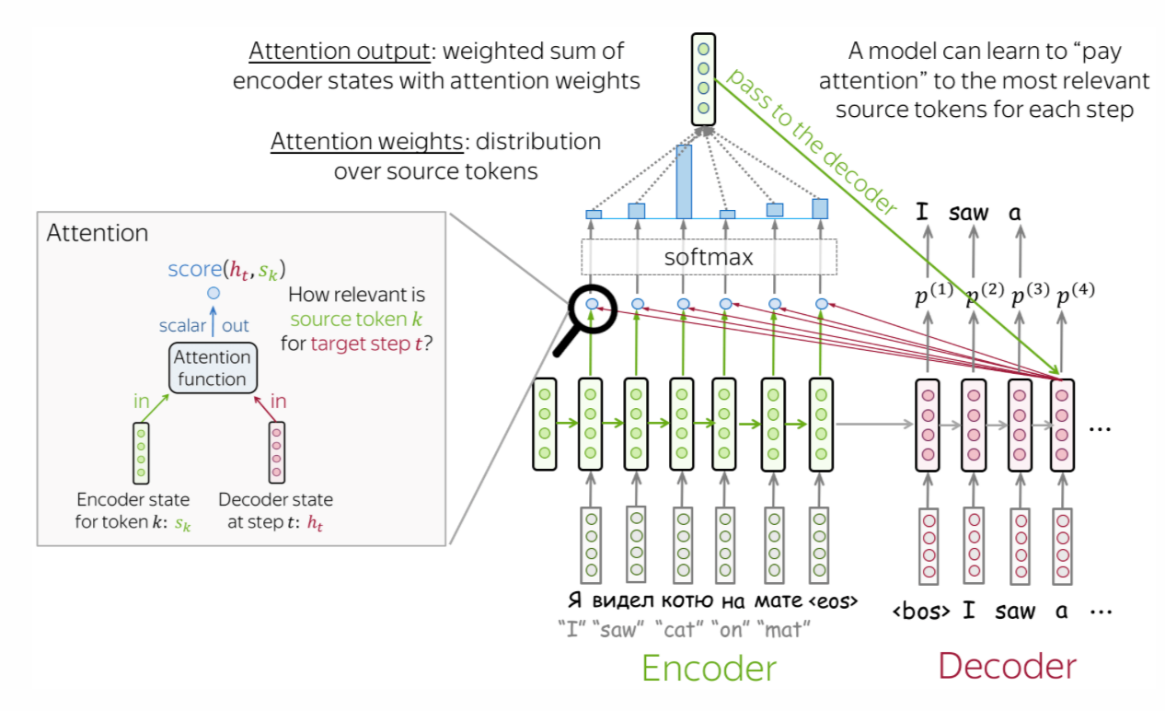
1. **Attention Mechanism in Sequence to Sequence Model:**

Attention mechanisms have become a crucial component in the design of advanced neural network architectures, particularly those dealing with sequence-to-sequence tasks such as language translation, speech recognition, and text summarization. The concept borrows from human cognitive attention, highlighting the model’s ability to focus on specific parts of the input data when performing a task, which enhances its performance and interpretability.

Definition of Attention in the Context of Neural Networks: *Attention is a technique that allows models to weigh the relevance of different inputs differently, focusing more on those aspects that are crucial for a task at a specific moment in the computation. It can be viewed as a mechanism of selectively concentrating on a few relevant things while ignoring others in large datasets or long input sequences.*

1. **How attentions works:**

This image illustrates the attention mechanism in a neural network, specifically in a sequence-to-sequence (seq2seq) model, which is commonly used in machine translation.



**Encoder**: The image's "Encoder" section at the bottom displays how the input sequence was processed. The encoder turns every word (or token) in the input language (like Russian) into a vector. After processing each word, the encoder's status is shown by each green box. These states are intended to convey the main ideas of the words as well as the sentence's context.

**Decoder**: The creation of the translated sequence in the target language (for instance, English) is displayed on the right side, labeled "Decoder." At each stage of creating the translation, the decoder's current condition is shown by a pink box.

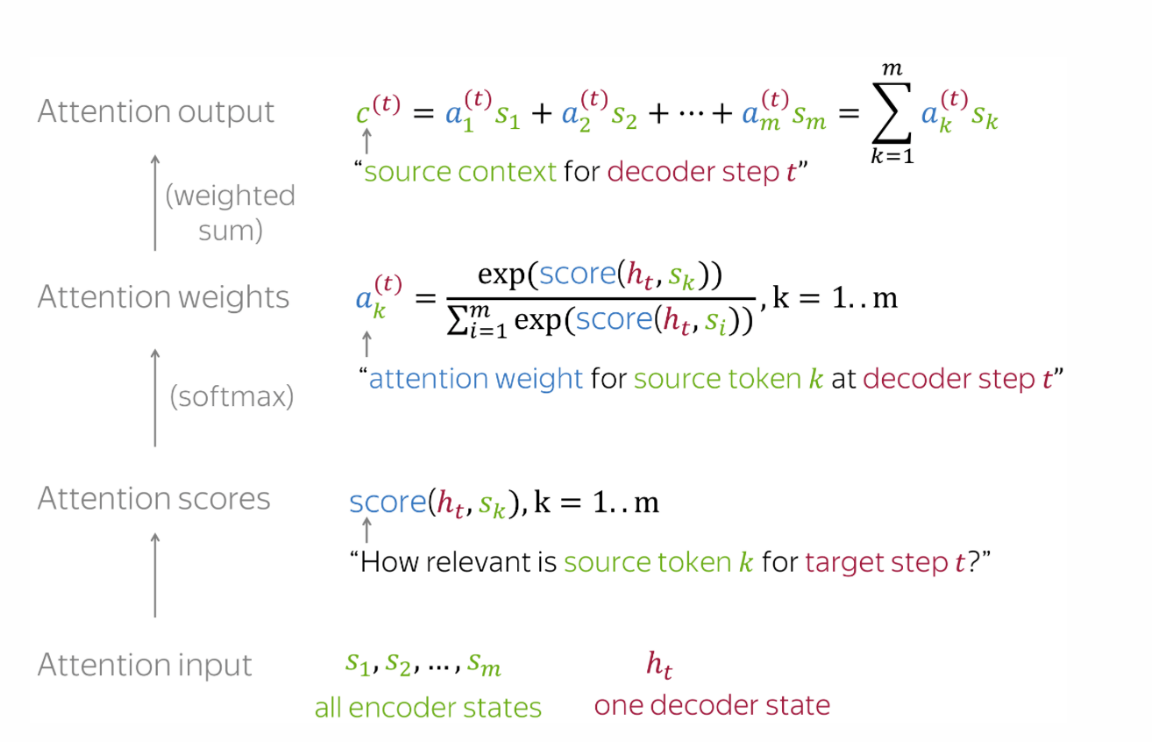
**Attention Mechanism**: The attention mechanism functions in the region in the center that is marked "Attention." The attention mechanism determines a score between the current decoder state and each of the encoder states for every word the decoder is attempting to output. When translating the current word, this score indicates how much attention should be paid to each word in the input sequence.

**Scores to weight**s: After that, the scores are normalized into a probability distribution called attention weights by running them through a softmax function. At the present stage of the decoding process, these weights establish how much "attention" or significance each input word should be given.

**Context for decoding**: The attention output is produced by combining the encoder states into a weighted sum using the attention weights. With an emphasis on the pertinent portions for the current translation step, this output is a vector that includes information from the input sequence.

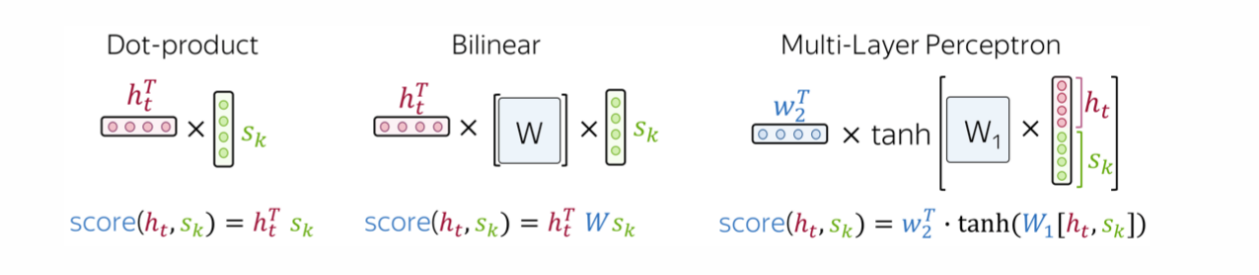
**Generating translation**: The following word in the output sequence is created by combining the attention output with the decoder's current state. Instead of depending on a single, fixed representation of the phrase, this method enables the decoder to "pay attention" to various sections of the input sentence as needed.

**Sequential Processing**: The attention mechanism recalculates the weights to adjust the emphasis as the decoder advances through the output sequence from one word to the next. When translating "I saw a cat," for example, the decoder may pay greater attention to the word "cat" in the input sequence while generating the word "cat" in the output sequence.



1. **How to compute Attention Score:**

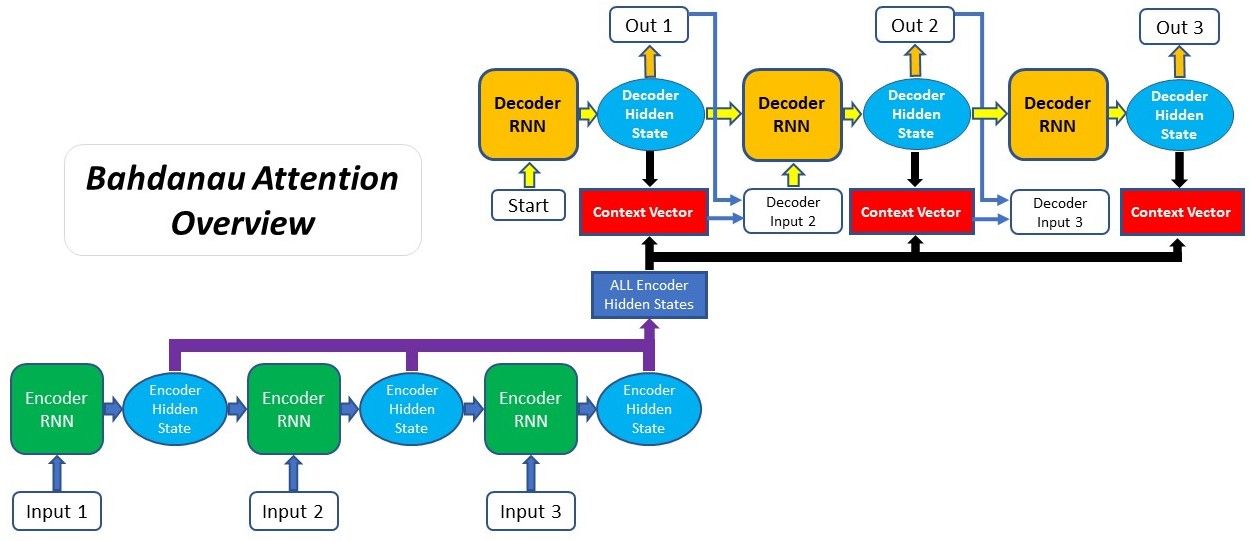
The most popular ways to compute attention scores are:



* Dot-product: The simplest method.
* Bilinear function (Luong attention): Used in the paper Effective Approaches to Attention-based Neural Machine Translation.
* Multi-layer perceptron (Bahdanau attention): The method proposed in the original paper.

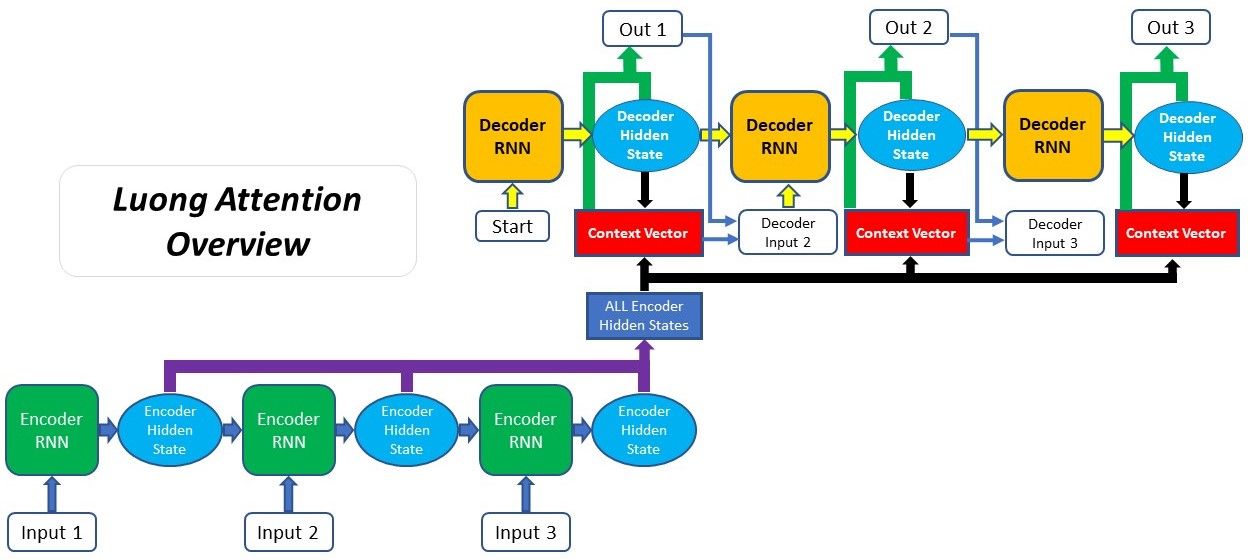
**Bahdanau Model:**

* The score is calculated by the additive attention using a feed-forward network with a single hidden layer. This technique applies a weighted sum with a non-linearity (usually tanh) after concatenating the encoder and decoder states.
* Two Recurrent Neural Networks (RNNs) are used in the encoder of the Bahdanau model; one reads the input sequence forward, while the other reads it backward. In languages where a word's meaning may be influenced by following words, this method enables the model to collect information from both past and future contexts in relation to the present token.
* To create a more thorough depiction of each token's context within the overall sequence, the model concatenates the forward and backward RNN states for each token in the input sequence. The attention mechanism then receives this concatenated state as input.



**Luong Model**: Minh-Thang Luong et al. developed the Luong attention technique, commonly referred to as "global attention," which enhances the performance of the Bahdanau et al. attention mechanism while simplifying it. The Luong attention model is distinguished by its unique method of applying attention in the decoder and determining attention scores. The Luong attention model may be summed up as follows:

* Differences from Bahdanau Attention - The primary difference between Luong attention and Bahdanau attention is in the calculation of the attention scores:
* Bahdanau (Additive Attention): Uses a multi-layer perceptron to compute the attention scores, which involves a feed-forward network with one hidden layer.
* Luong (Global Attention): Simplifies this computation by using various scoring functions that can be as simple as a dot product between the encoder and decoder states.
* Attention score computation: Luong proposed several alternatives for computing the alignment scores, such as:
* Dot Product: Directly computes the dot product of the decoder’s current hidden state and all the encoder’s hidden states.
* General: Computes the score as the decoder’s current hidden state multiplied by a learned weight matrix, and then the encoder’s hidden states.
* Concatenation: Similar to Bahdanau, but typically uses a simpler architecture.



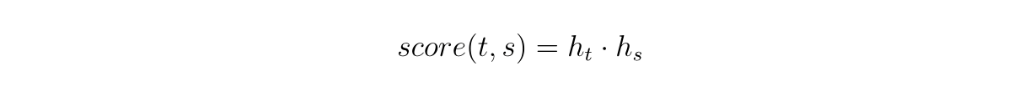
1. **Attention Mechanism in LSTM Model:**

Since the advent of models like Transformers, which mostly rely on self-attention to interpret input data, the attention mechanism has been a key component of many state-of-the-art outcomes in a variety of NLP tasks.

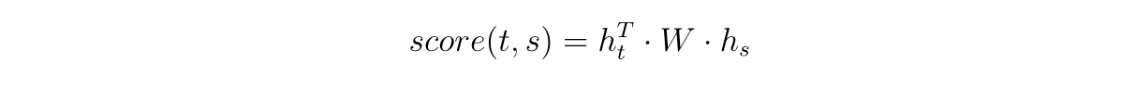
Because of their superior ability to capture long-range dependencies, attention mechanisms may be linked to both transformers and LSTM and GRU networks.

Implementing attention mechanism for LSTM:

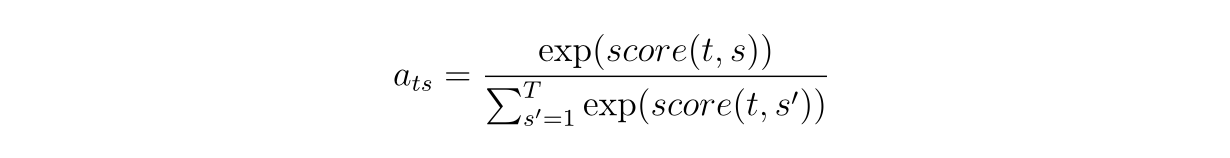
* Define the LSTM: An LSTM encoder-decoder structure will be used. After processing the input sequence, the encoder LSTM generates a series of hidden states. The forward and backward cell states of the LSTM at time s are usually concatenated to form each hidden state hₛ.
* Compute the Attention Scores: Use all of the encoder hidden states to calculate attention ratings for each hidden state hₜ of the decoder LSTM. A frequently employed technique is the dot-product scoring function:



Alternatively, other scoring functions like multiplicative or additive attention can be used. The multiplicative approach, for example, employs a weight matrix *W*:

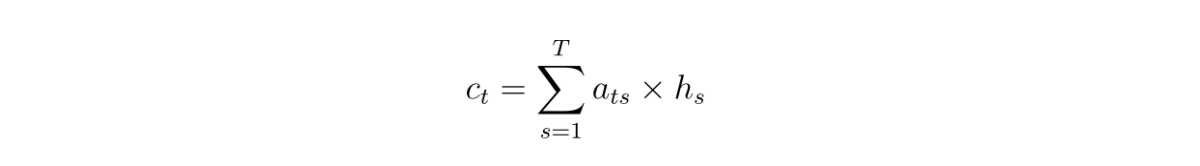


* Compute the Attention Weights: Using the softmax function, normalize the scores to create a probability distribution:



Here, aₜₛ represents the attention weight for encoder hidden state hₛ​ when decoding at time step t.

* Compute the Context Vector: Calculate the context vector for the decoder time step *t* as a weighted sum of the encoder hidden states:



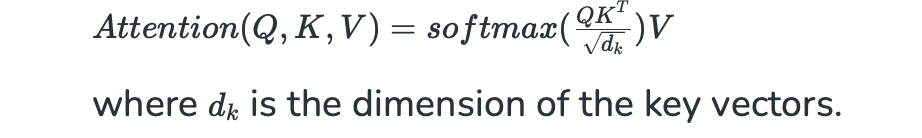
* Concatenate or Combine the Context Vector: The hidden state hₜ of the decoder may then be combined with the context vector cₜ. This usually entails concatenating cₜ​ and hₜ​, which is then fed into a dense layer to provide either the output prediction for that time step or the input for the subsequent LSTM layer.
* **Training**: Backpropagation through the attention mechanism and the LSTM is used to train the model. In order to generate correct outputs, the model must learn where to concentrate its attention within the input sequence.
* **Decoding**: Use techniques like greedy decoding and beam search during inference to generate the output sequence. At each stage of the decoding process, the attention mechanism will dynamically focus on various segments of the input sequence.

1. **Attention Mechanism in Transformer Model:**

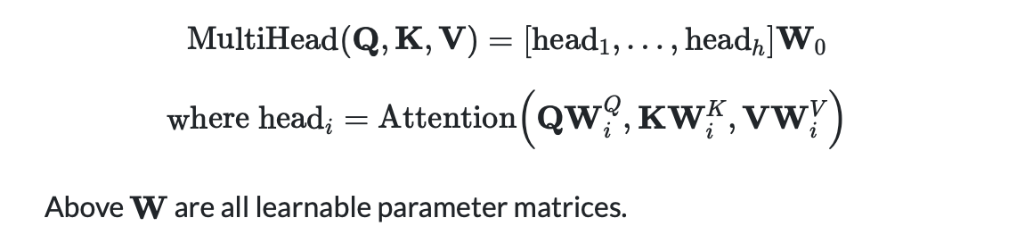
Unlike traditional models like LSTM, Transformer does not rely on sequentiality but takes advantage of Attention's parallel processing capabilities to capture the relationship between words in a sentence more flexibly.

The Attention mechanism in Transformer is called Multi-Head Self-Attention. It allows the model to learn many different types of relationships between words in a sentence, thereby significantly improving the model's performance.

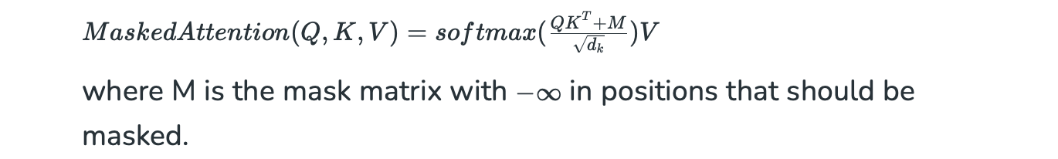
* **Scaled Dot-Product Attention**: The Scale Dot-Product is the core component of the Transformer's attention mechanism is attentiveness. It consists of three primary parts: values (V), keys (K), and queries (Q). The dot product of the query and key vectors, scaled by the square root of the key vectors' size, is how the attention score is calculated. A weighted sum of the value vectors is then calculated using the attention weights that are obtained by passing this score through a softmax function.



* **Multi-Head Attention**: The model's capacity to concentrate on several segments of the input sequence at the same time is improved by multi-head attention. There are several attention heads involved, and each has a unique set of value, key, and query matrices. The final output is created by concatenating and linearly transforming the outputs of these heads. This enables the model to identify various relationships and characteristics in the input sequence.



* **Self-Attention**: When determining a word's representation, the model may take into account various places within the same sequence thanks to self-attention, sometimes referred to as intra-attention. Self-attention is used in both the encoder and decoder levels of the Transformer. It makes it possible for the model to identify long-range linkages and dependencies in the input sequence.
* **Encoder-Decoder Attention**: Decoder-Encoder Attention in the Transformer's decoder layers make use of attention, sometimes referred to as cross-attention. When creating each word of the output sequence, it enables the decoder to concentrate on pertinent portions of the input sequence (encoded by the encoder). By guaranteeing that the decoder has access to the complete input sequence, this kind of attention aids in the production of translations that are more precise and suitable for the context.
* **Causal or Masked Self-Attention**: To make sure that the forecast for a particular location only depends on the known outputs at places preceding it, the decoder employs Causal or Masked Self-Attention. In tasks like language modeling, where future tokens shouldn't be seen during training, this is essential. The model is prevented from looking forward by masking off the attention ratings for future coins.



1. **Attention Mechanism in GPT (Generative Pretrained Transformer) Model:**

The transformer architecture, upon which GPT is built, was first presented in the 2017 publication "Attention is All You Need" by Vaswani et al. The transformer's main concept is the employment of self-attention mechanisms, which, in contrast to conventional approaches that process words in sequential order, process words in relation to every other word in a phrase. This gives the model a more sophisticated grasp of language by enabling it to consider the significance of every word regardless of where it appears in the sentence. GPT may generate fresh material since it is a generative model. GPT may produce logical and contextually appropriate continuations when given a prompt or a sentence fragment. Because of this, it is quite helpful for tasks like writing creatively, developing textual material, and even mimicking dialogue.

The transformer architecture, which is the foundation of GPT models, is made up of feedforward neural networks and layers of self-attention processes.

Important elements of this architecture consist of:

* **Self-Attention System**: This allows the model to assess the meaning of each word in relation to the entire input sequence. It enables the model to understand word dependencies and connections, which is necessary for generating material that makes sense and is appropriate for its context.
* **Layer normalization and residual connections**: These features improve network convergence and help with training stability by lowering issues like exploding and vanishing gradients.
* **Feedforward Neural Networks**: These networks offer an additional layer of abstraction and learning capabilities while processing the attention mechanism's output. They are situated in between levels of self-attention.

At the forefront of this change is the GPT (Generative Pretrained Transformer) model developed by OpenAI. This article delves into the attention mechanism, specifically how it’s employed in GPT models. The GPT models, like their transformer counterparts, employ a mechanism known as “scaled dot-product attention.” The architecture of the GPT model is essentially a stack of identical layers, each comprising two sub-layers: a multi-head self-attention mechanism, and a position wise fully connected feed-forward network.

Consider a scenario where:

* The model processes a sentence with n words (sequence length).
* Each word is represented by a d dimensional vector (d being the model's hidden size).
* We use h attention heads.
* Dimension of keys, queries, and values is d\_k.

The input matrix X has dimensions n x d. The weight matrices W\_Q, W\_K, W\_V are of dimensions d x d\_k, leading to Q, K, and V matrices of size n x d\_k.

The attention scores QKᵀ result in an n x n matrix representing attention scores for each word relative to every other word. Applying softmax to these scores gives the attention weights α, also an n x n matrix.

The output of the attention mechanism for each head (αV) is an n x d\_k matrix. The concatenated output of all heads is n x (h\*d\_k). Finally, after a linear projection by W\_O (of dimension h\*d\_k x d), we obtain the output of the multi-head attention layer, which is n x d.

→ The attention mechanism, a fundamental part of GPT models, has proven instrumental in advancing NLP. Attention has produced previously unheard-of coherence and contextually appropriate outputs from models by enabling them to dynamically focus on various input components and developing a more sophisticated knowledge of context. To appreciate and operate with such sophisticated NLP models, one must comprehend the mathematics and underlying concepts of attention.

# REFERENCE