

CS409 : Neural Networks (Semester II - 2021/22)

Unit 5: Recurrent Neural Networks (2)

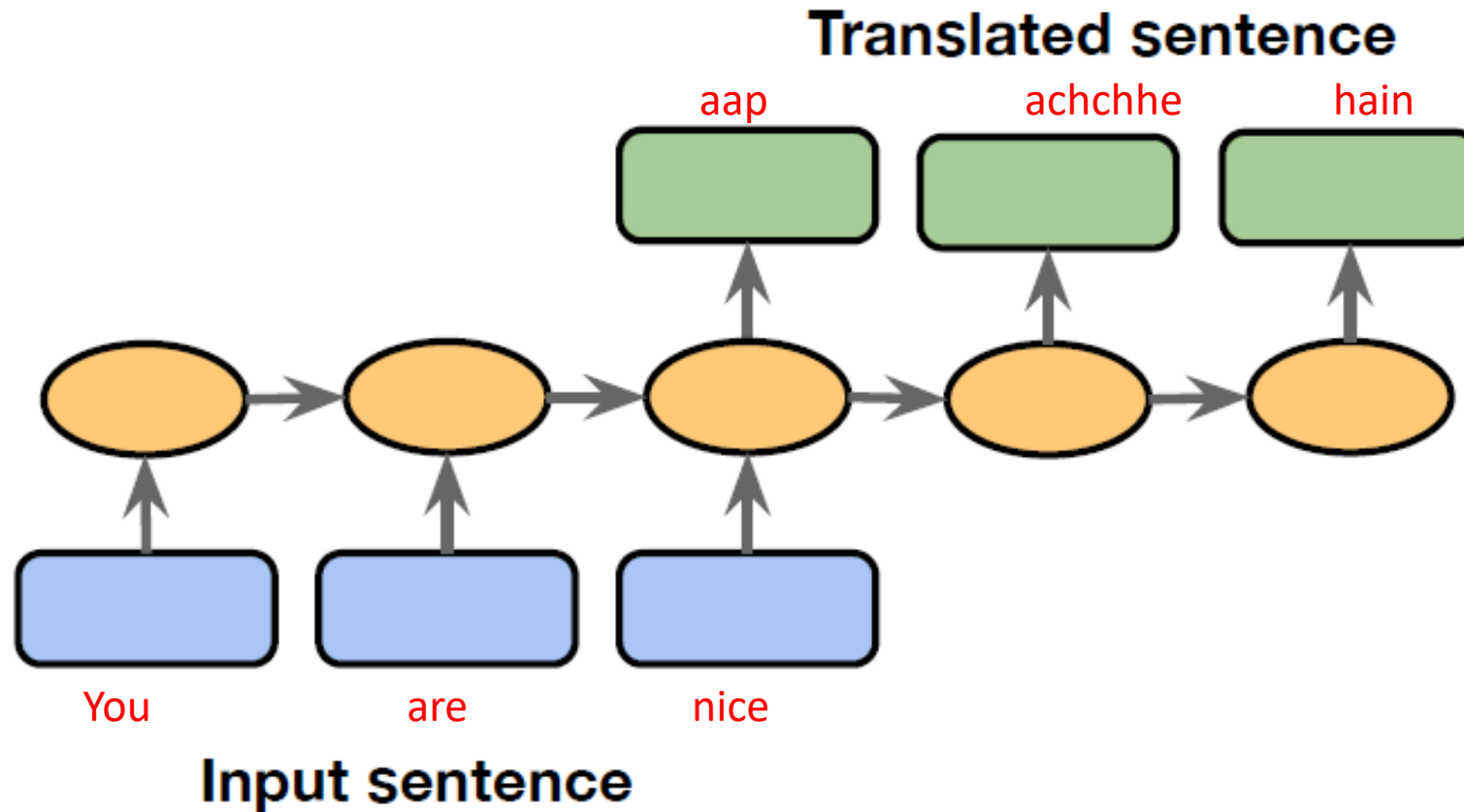
Dr. Ruwan Nawarathna
Department of Statistics & Computer Science
Faculty of Science
University of Peradeniya



Sequence Modeling Applications

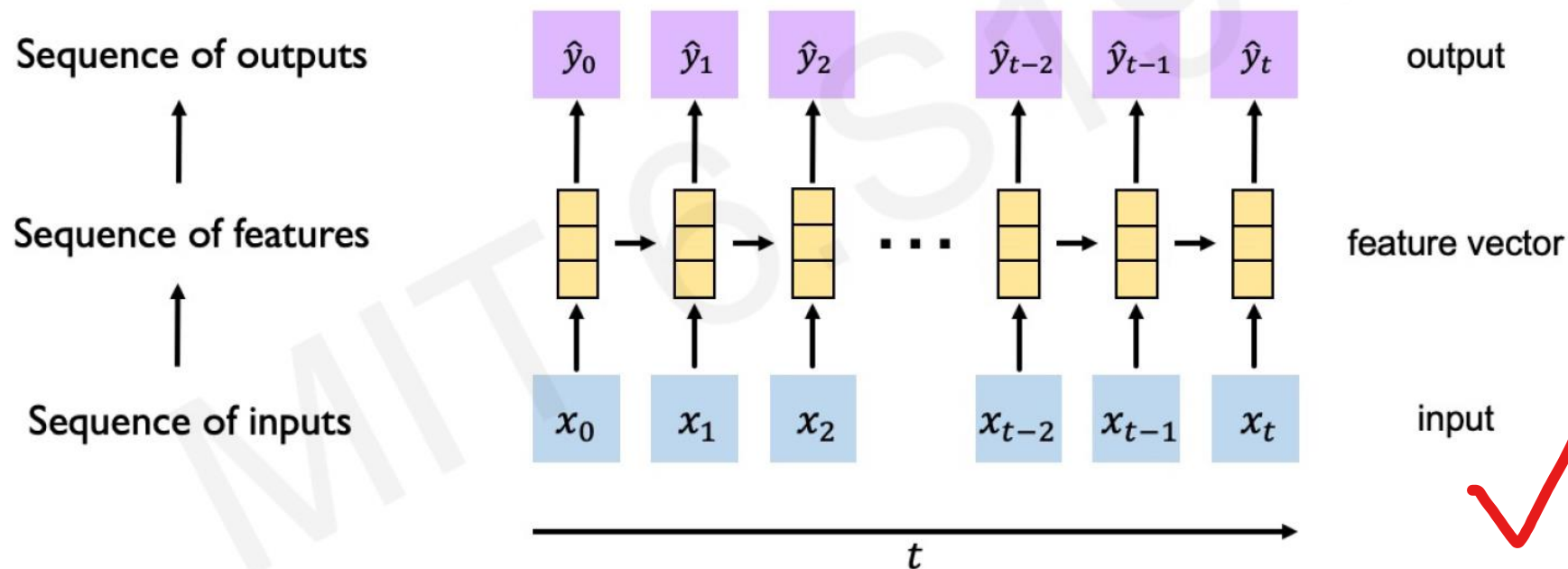
- Sequence to symbol
 - Sentiment analysis
 - Best movie ever – Positive
- Sequence to Sequence
 - NLP: Named Entity Recognition
 - Input: Jim bought 300 shares of Acme Corp. in 2006
 - NER: [Jim]Person bought 300 shares of [Acme Corp.]Organization in [2006]Time
 - Machine Translation
 - Echte dicke kiste - Awesome sauce

Encoder-Decoder Sequence to Sequence Models



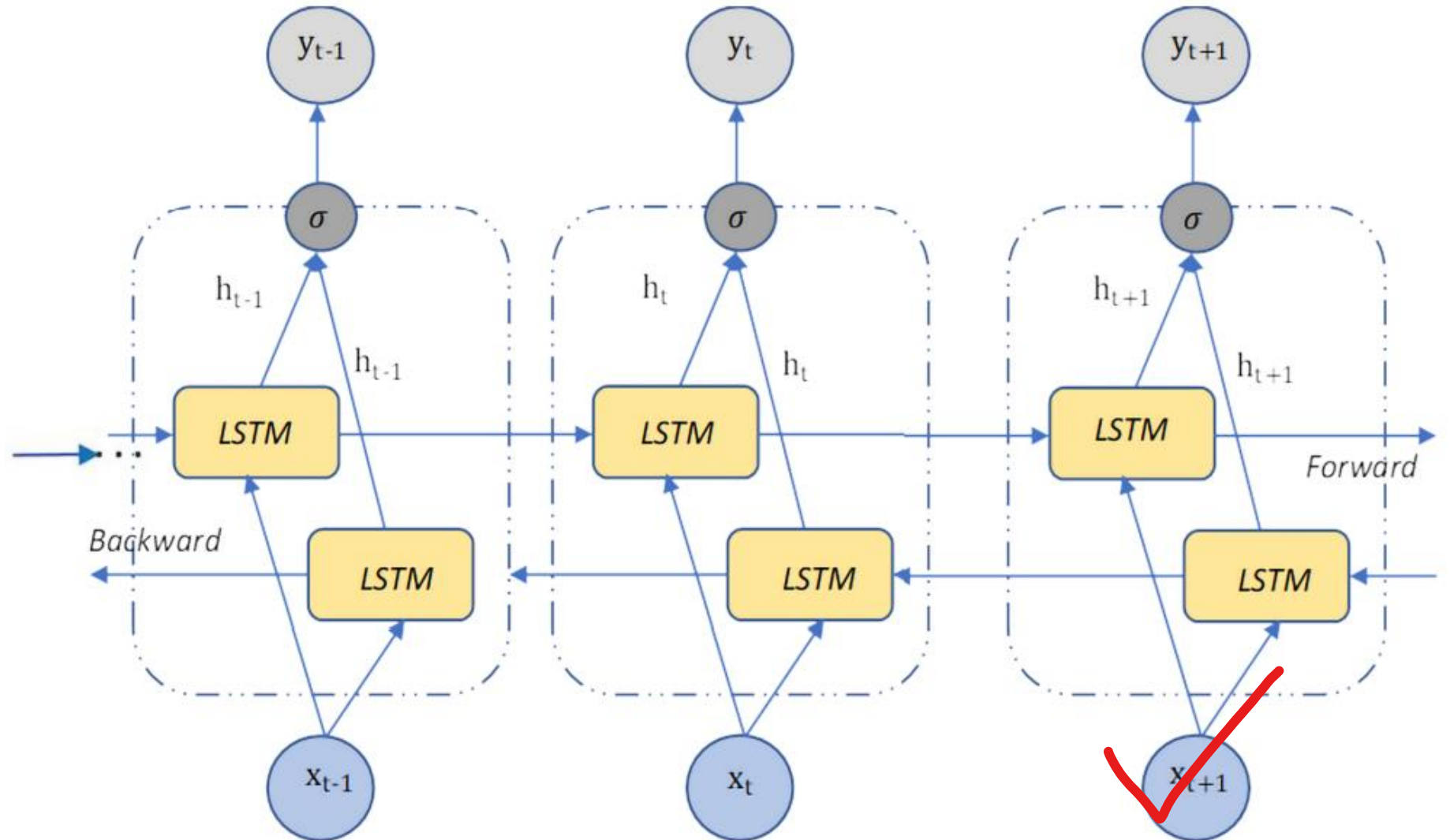
Limitations of RNNs (LSTMs)

- Encoding bottleneck
- Slow, no parallelization
- Not long memory



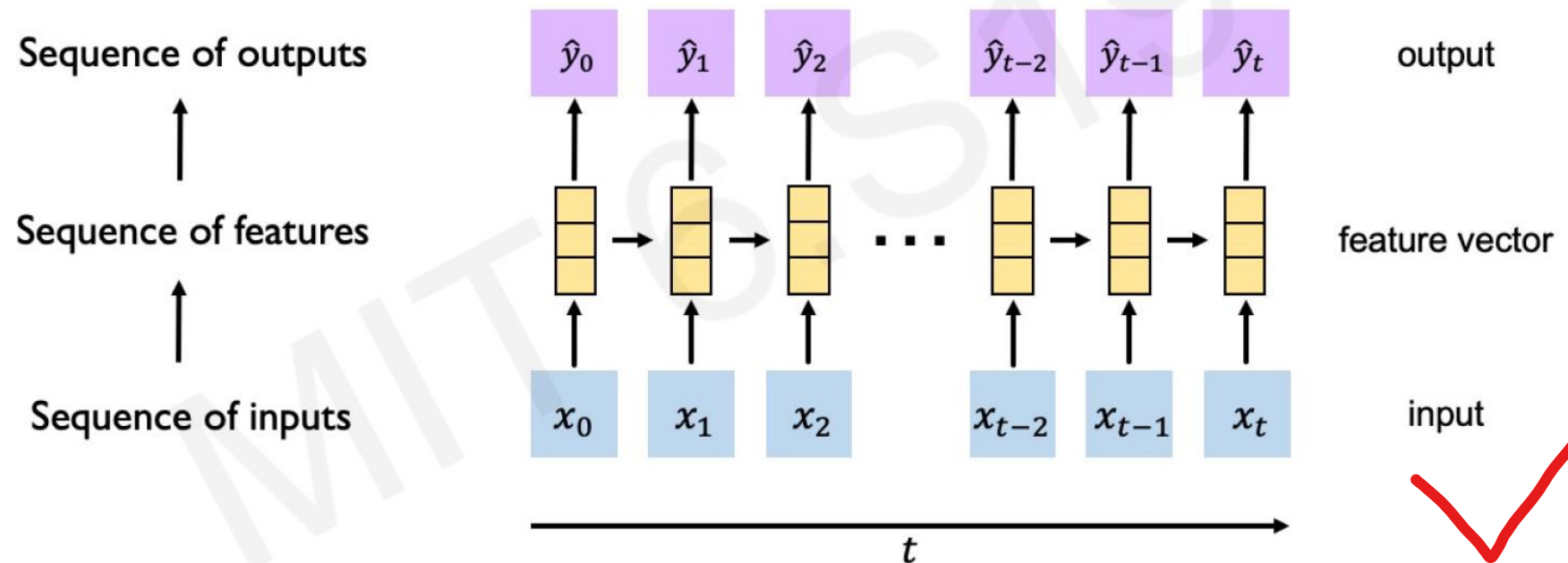
Bi-directional LSTM

Bidirectional LSTM or BiLSTM contains two LSTM layers, one for processing input in the forward direction and the other for processing in the backward direction.



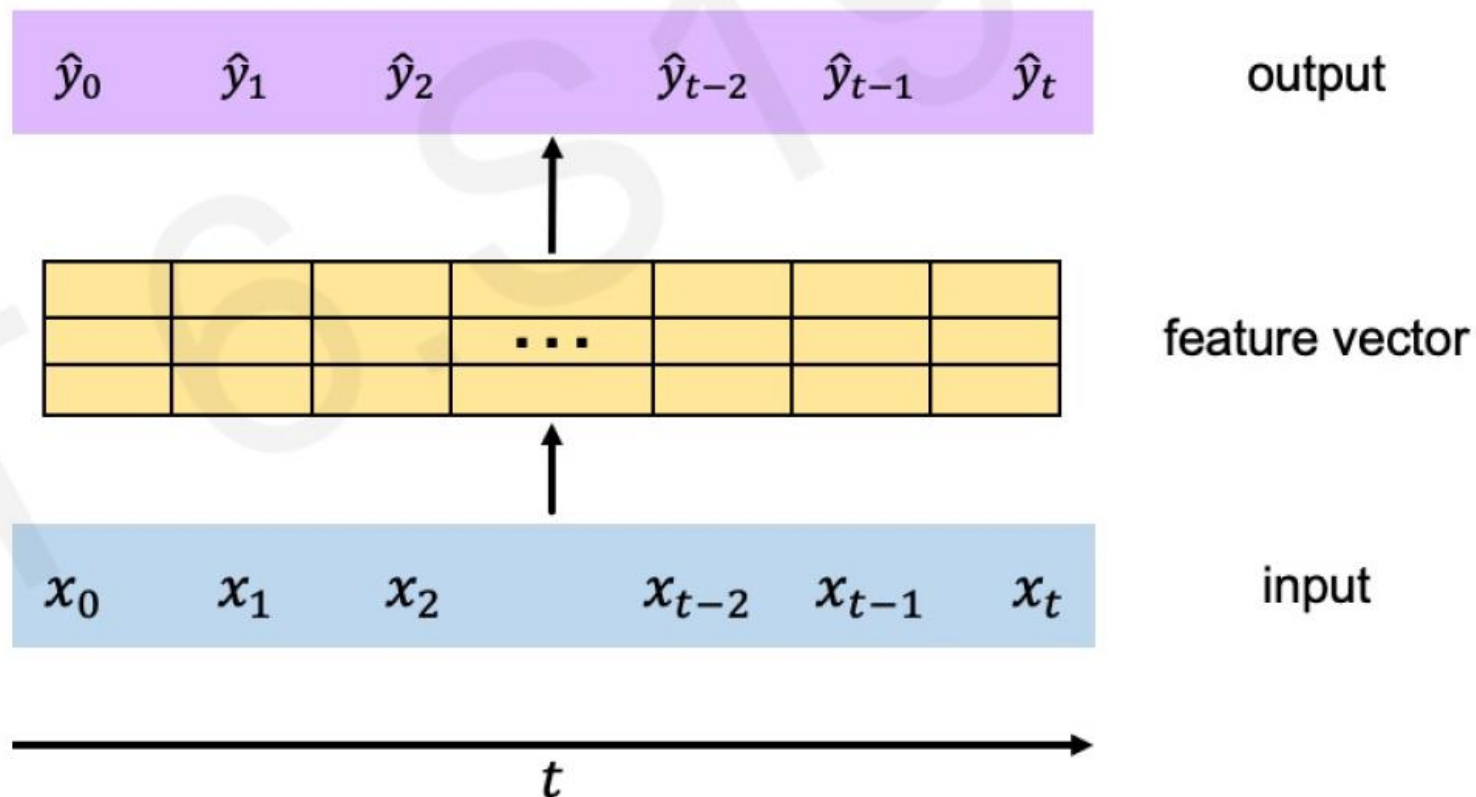
Desired Capabilities of LSTMs/Bi-LSTMs

- Continuous stream
- Parallelization
- Long memory






Desired Capabilities of LSTMs/Bi-LSTMs

- Can we eliminate the need for recurrence entirely?



Attention

- The attention mechanism in NLP is one of the most valuable breakthroughs in Deep Learning research in the last decade. 
- It has spawned the rise of so many recent breakthroughs in natural language processing (NLP), including the Transformer architecture. 
- The attention mechanism has changed the way we work with deep learning algorithms 

Attention

- Assign attention weight to each word, to know how much "attention" the model should pay to each word
(i.e., for each word, the network learns a "context")
- **Attention all you need**
 - Identify which parts to attend to
 - Extract the features with high attention



Attention all you need

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Lukasz Kaiser*
Google Brain
lukaszkaiser@google.com

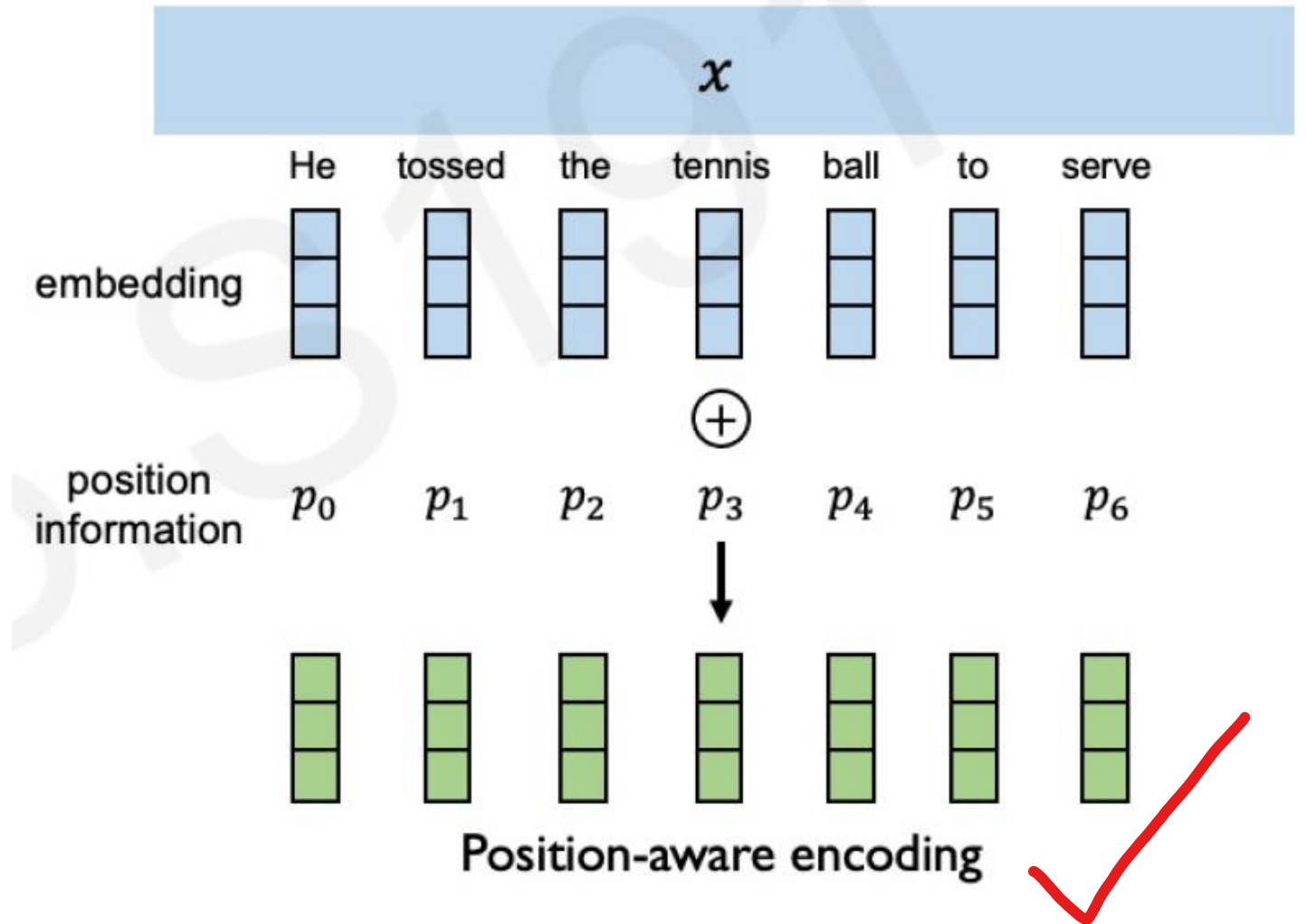
Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).



Attention

1. Encode position information
2. Extract **query**, **key**, **value** for search
3. Computer attention weighting
4. Extract features with high attention



Self-Attention Mechanism

- Previous basic version did not involve any learnable parameters.
 - so not very useful for learning a language model
- We are now adding 3 trainable weight matrices that are multiplied with the input sequence embeddings (x_i 's)

$$\text{query} = \mathbf{W}^q \mathbf{x}_i$$

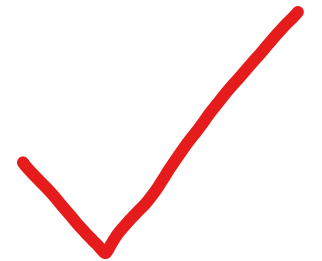
$$\text{key} = \mathbf{W}^k \mathbf{x}_i$$

$$\text{value} = \mathbf{W}^v \mathbf{x}_i$$

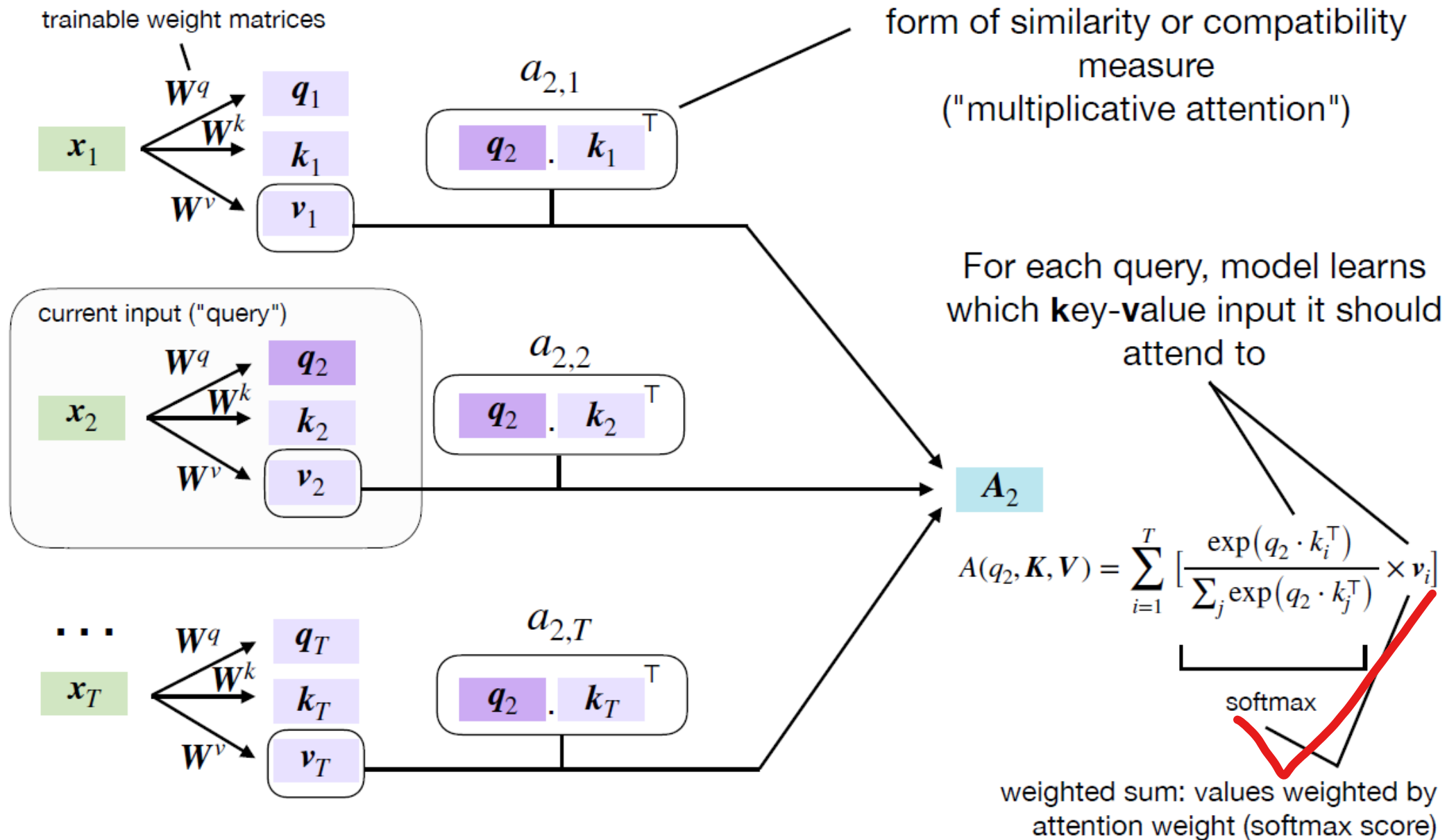


Self-Attention Mechanism - Query, Key, Value

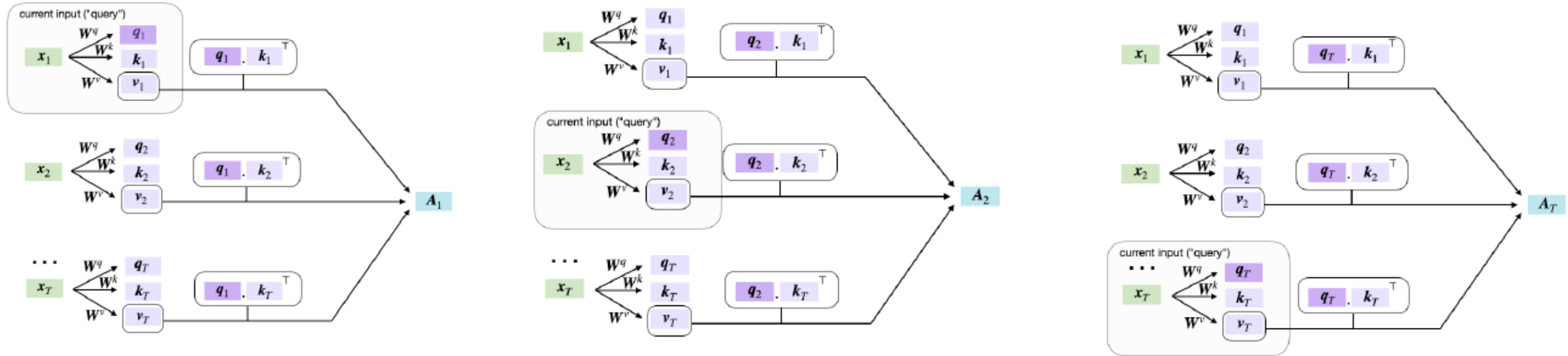
- Every input vector x_i is used in 3 ways:
 - Compared to every other vector to compute attention weights for its own output y_i (query)
 - Compared to every other vector to compute attention weight w_{ij} for output y_j (key)
 - Summed with other vectors to form the result of the attention weighted sum (value)



Self-Attention Mechanism



Self-Attention Mechanism



Attention score matrix: $A = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix}$

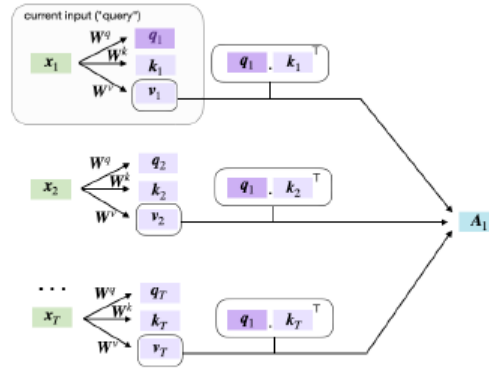


Self-Attention Mechanism (Scaled Dot Product Attention)

d_e = embedding size

T = input sequence size

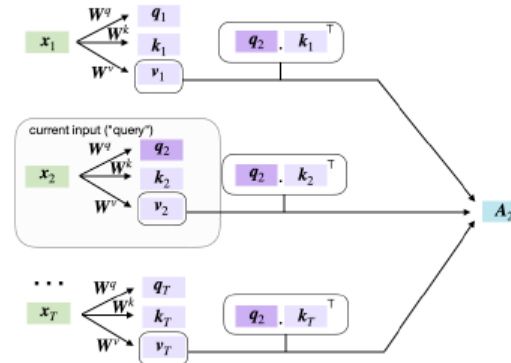
$$\mathbf{x} \in \mathbb{R}^{T \times d_e}$$



$$\mathbf{Q} \in \mathbb{R}^{T \times d_q}$$

$$\mathbf{K} \in \mathbb{R}^{T \times d_k}$$

$$\mathbf{V} \in \mathbb{R}^{T \times d_v}$$

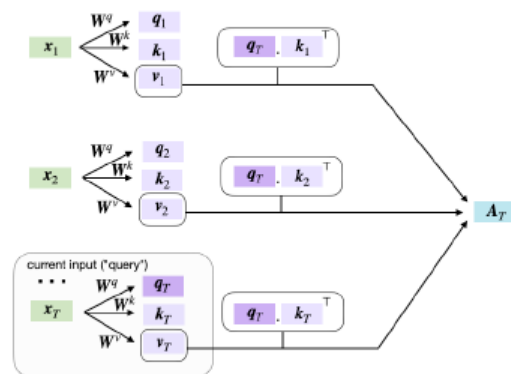


"attention matrix"

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

$T \times T$

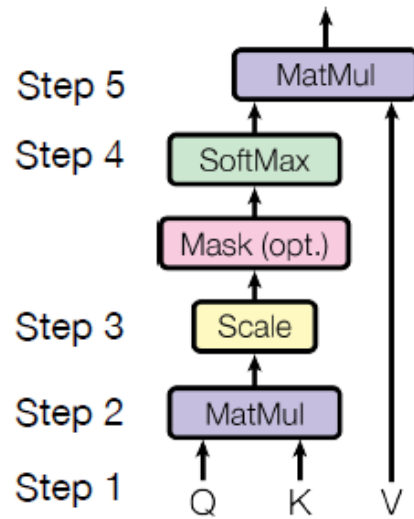
$T \times d_v$



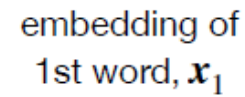
"attention-based embedding"

Scaled Dot Product Attention

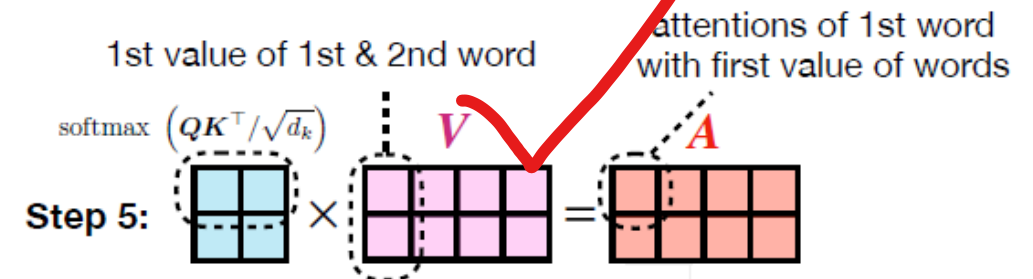
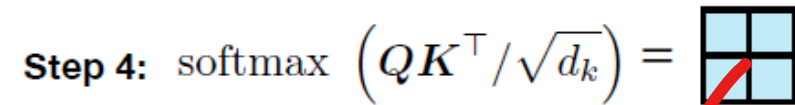
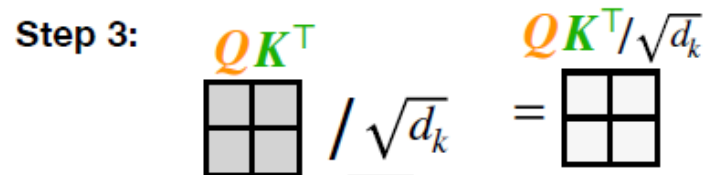
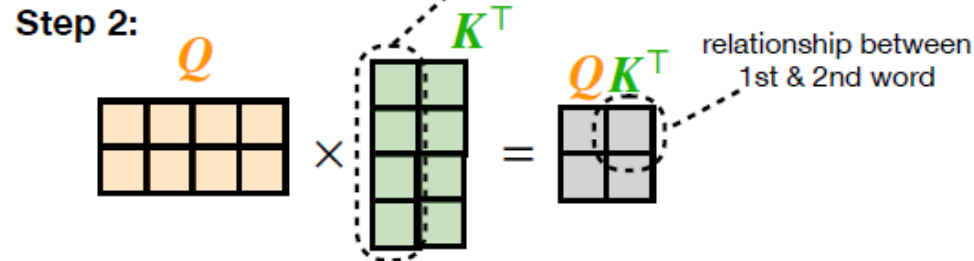
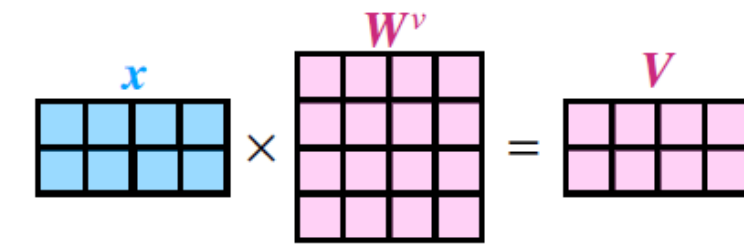
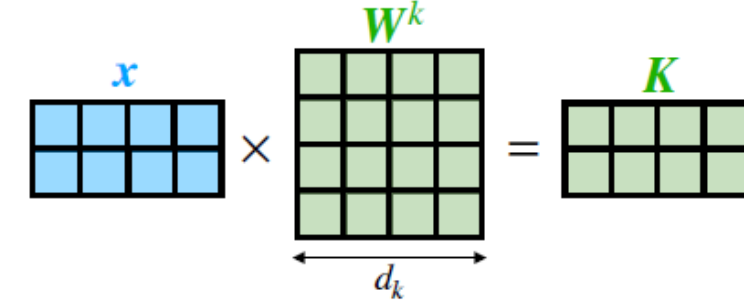
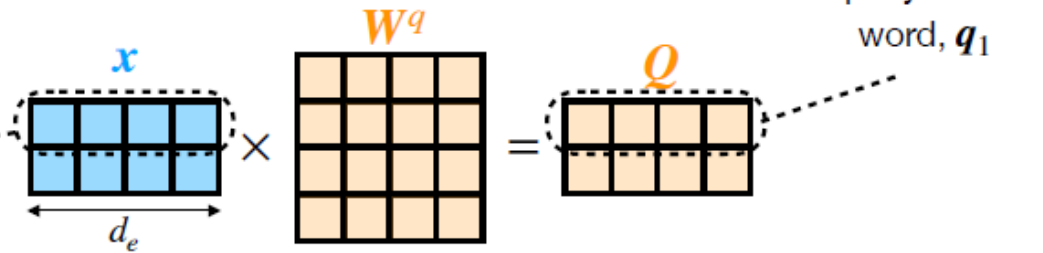
Scaled Dot-Product Attention



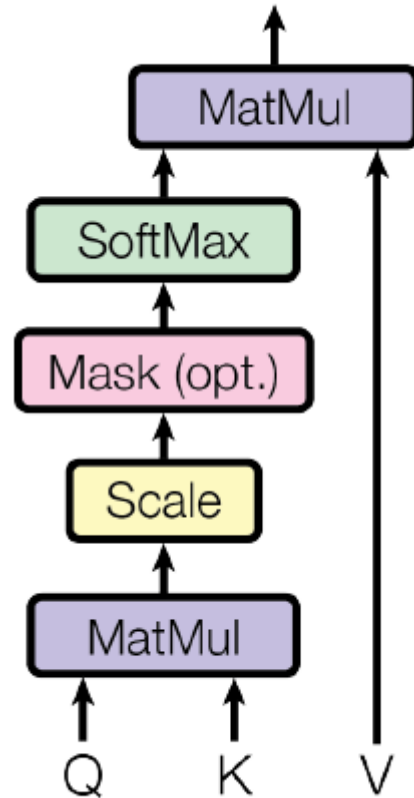
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I., 2017. Attention Is All You Need.



Step 1:



Self-Attention Mechanism (Scaled Dot Product Attention)



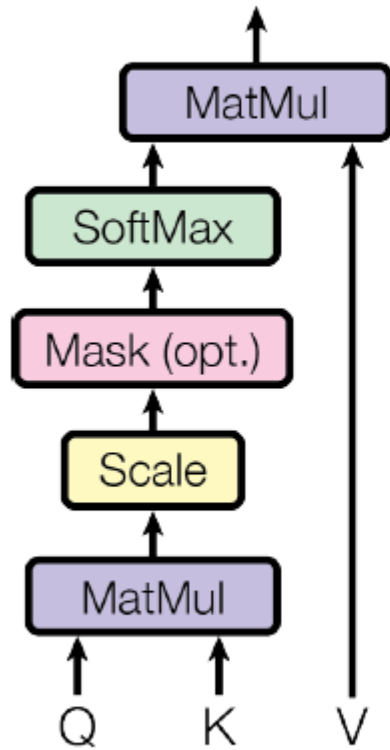
Multi-Head Attention

- Apply self-attention multiple times in parallel (similar to multiple kernels for channels in CNNs)
- For each head (self-attention layer), use different W^q, W^k, W^v , then concatenate the results, $A_{(i)}$
- 8 attention heads in the original transformer, i.e.,
 $W_{(1)}^q, W_{(1)}^k, W_{(1)}^v \dots W_{(8)}^q, W_{(8)}^k, W_{(8)}^v$
- Allows attending to different parts in the sequence differently

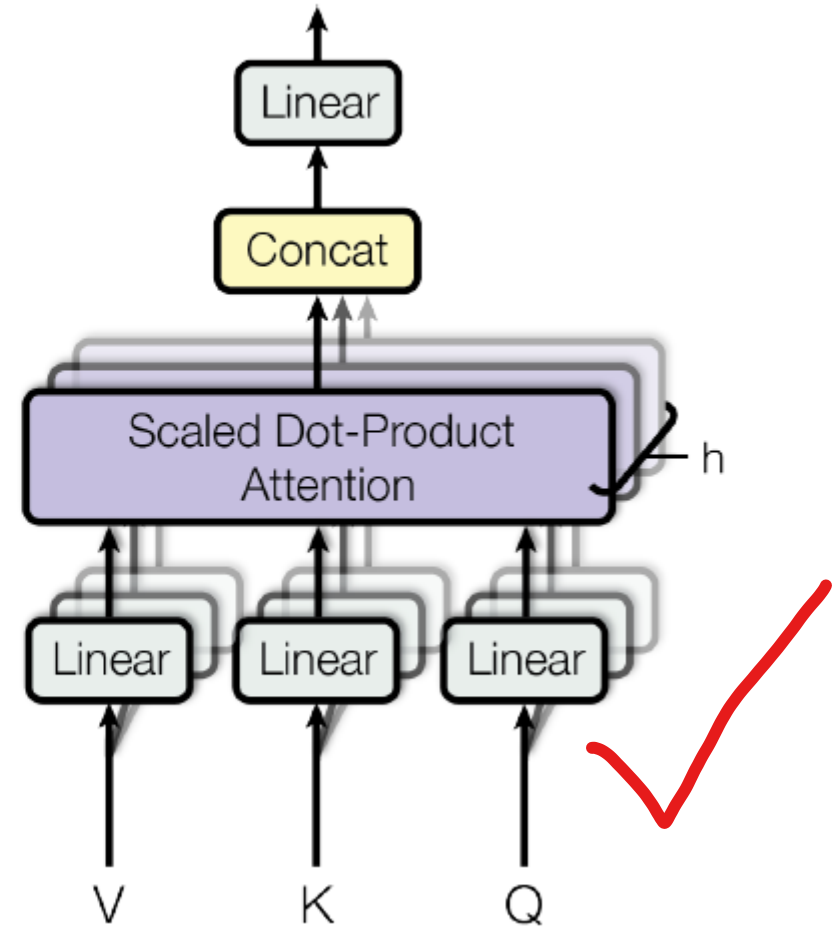


Multi-Head Attention

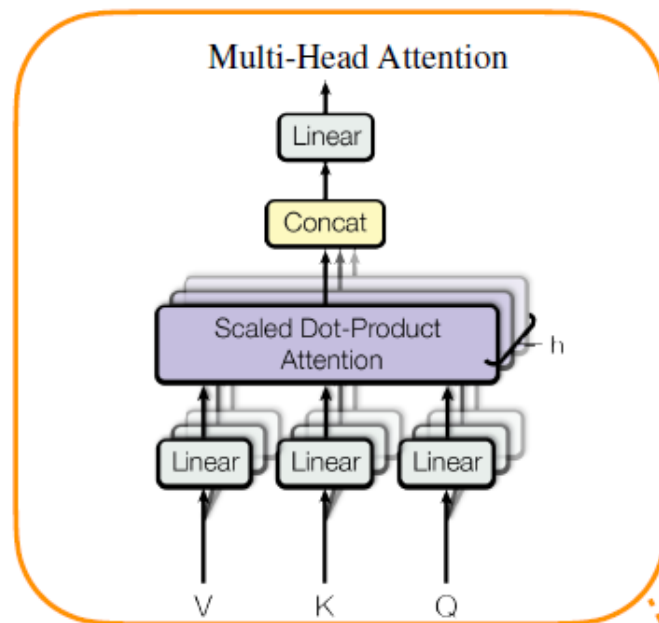
Scaled Dot-Product Attention



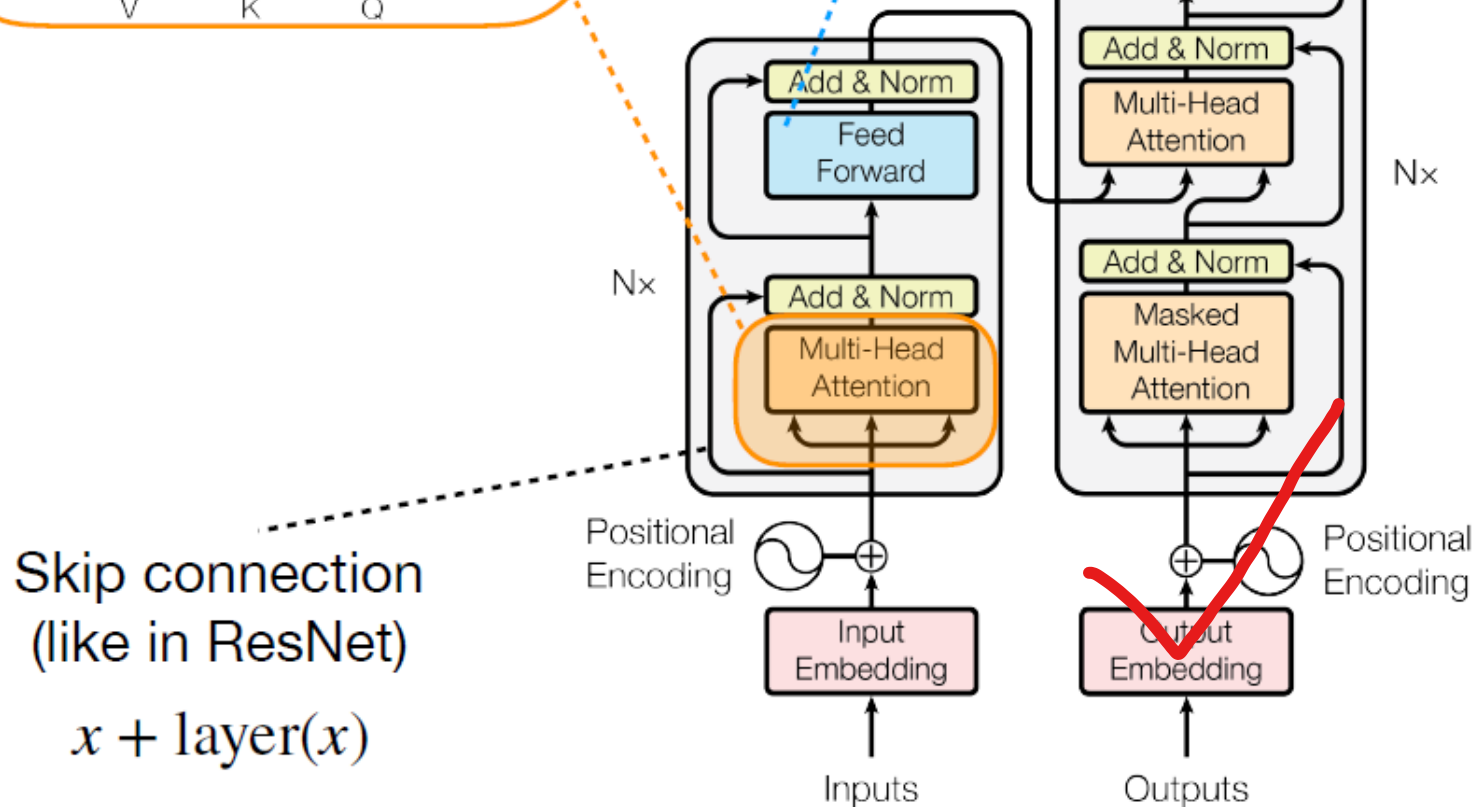
Multi-Head Attention



Transformer Models



Multilayer Perceptrons



Transformer Models

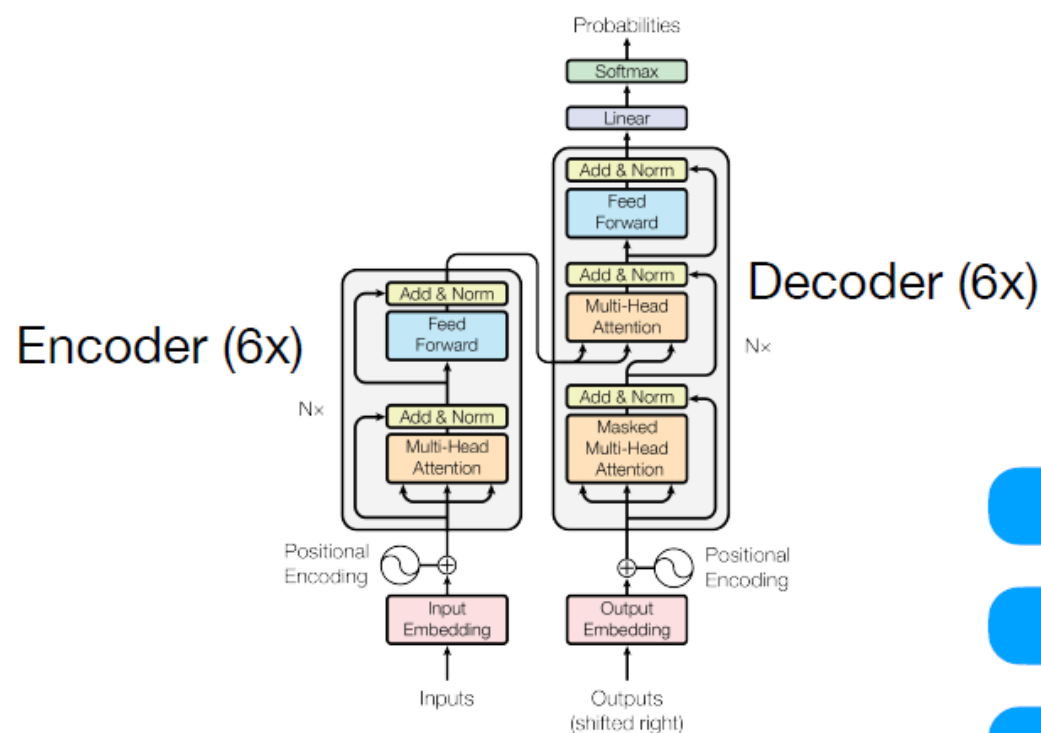
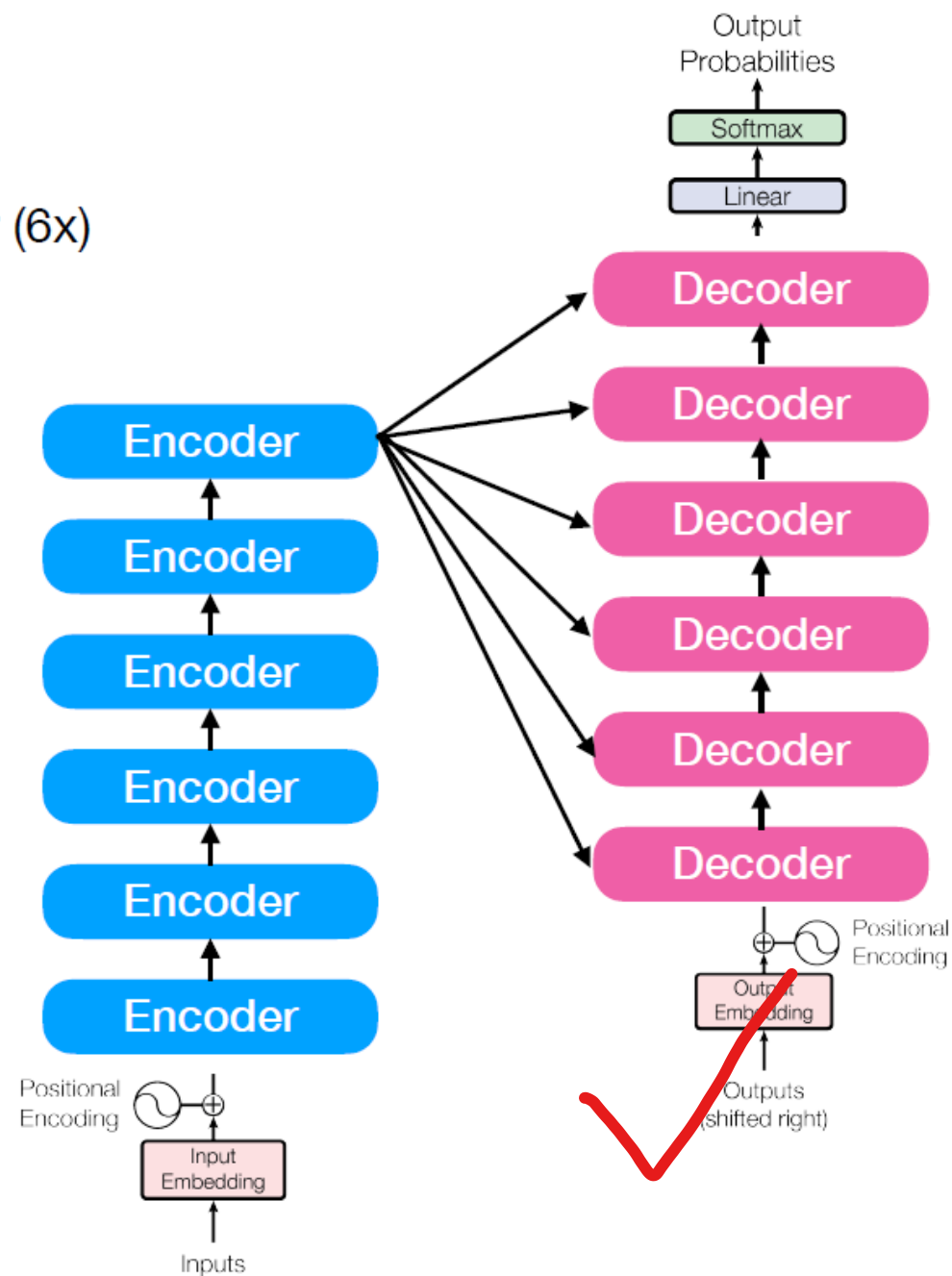


Figure 1: The Transformer - model architecture.

Same structure and dimension in/out for each encoder & decoder



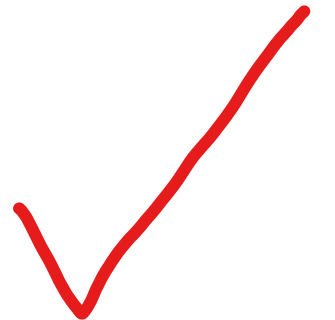
Transformer Models

- Pre-training on large unlabeled datasets
- Training for downstream-tasks on labeled data
 - fine-tuning approach
 - feature-based approach



Transfer Learning

- Transfer learning is a supervised learning method that aids construction of new models using pre-trained weights of previously constructed and fine-tuned models.
- Take a model trained on a large dataset and transfer its knowledge to a smaller dataset.



Pre-trained Models in NLP

- Pre-trained models (PTMs) for natural language processing (NLP) are deep learning models, such as transformers, that have been trained on large datasets to perform specific NLP tasks.
- By training on extensive corpora, PTMs can learn universal language representations, which are useful for various downstream NLP tasks such as text summarization, named entity recognition, sentiment analysis, part-of-speech tagging, language translation, sentiment analysis, text generation, information retrieval, text clustering, and many more.
- This eliminates the need to train a new model from scratch each time.



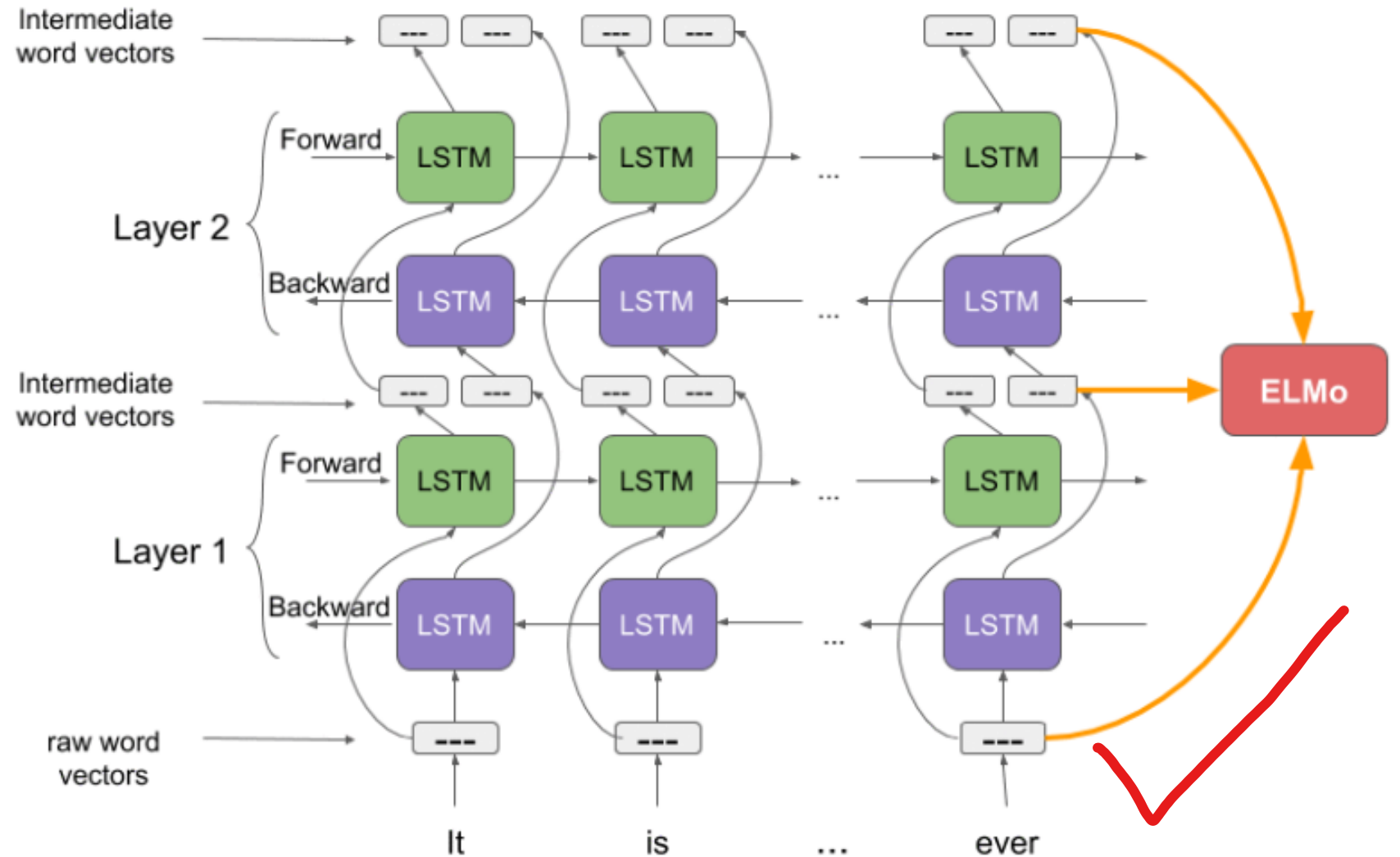
Popular Pre-trained Models in NLP

- GPT-4 (Generative Pre-trained Transformer 4)
- GPT-3 (Generative Pre-trained Transformer 3)
- BERT (Bidirectional Encoder Representations from Transformers)
- RoBERTa (Robustly Optimized BERT Pretraining Approach)
- T5 (Text-to-Text Transfer Transformer)
- XLNet (eXtreme Multi-task Learning Network)
- ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately)
- DistilBERT (Distilled BERT)
- ULMFiT (Universal Language Model Fine-tuning)
- ELMO (Embeddings from Language Models) – Word Embedding

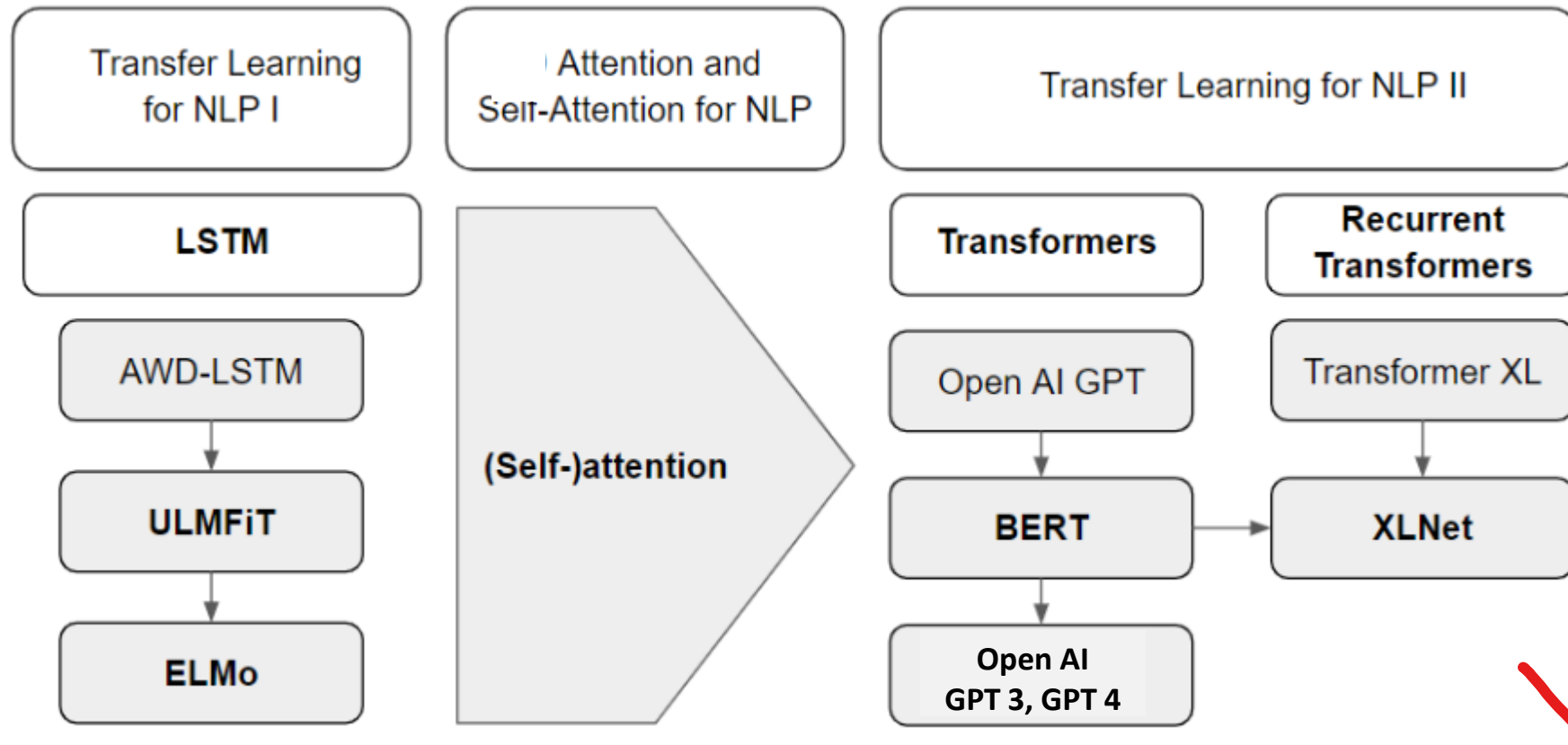


ELMO Word Embedding

- Uses two BiLSTMs



Overview of Important NLP Models



Example Application



Named Entity Recognition (NER)

- Named-entity recognition (NER) (also known as entity identification, entity chunking, and entity extraction) is a sub-task of information extraction that seeks to locate and classify named entities mentioned in unstructured text into pre-defined categories such as person names, organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc.



NER Example : Disease Named Entity Recognition (DNER)

Introduction

- Identifying the boundaries of a disease mentioned.
- Sources:
 - *Biomedical Literature (e.g., Medline Database) – Our focus*
 - *Online Health forums*
 - *General Social Media Platforms*
- An excerpt from a publication.

condition of the patients was graded according to the stages of disease defined by the Global Initiative for **Chronic Obstructive Lung Disease** (GOLD).¹² After the baseline visit, patients were followed for a total of seven visits: at 3 months, at 6 months,



Disease Named Entity Recognition (DNER)

The Need for an Automated DNER

- Health-related digital data is overwhelming; limits manual processing.
- Enable vigilant health monitoring and alert relevant parties when a novel disease entity is reported.
- Able to identify descriptive disease mentions.
- Helps information extraction related to diseases.
- Improve search results by better indexing the disease terms.



Disease Named Entity Recognition (DNER)

Methodology

- DNER is defined as a sequence labeling problem.
- B-I-O Scheme is applied (B-I-O : Begin-Inside-Outside).
- An example of a DNER task.

Input text:	new	diagnosis	of	prostate	cancer
Tagging results:	O	O	O	B-disease	I-disease

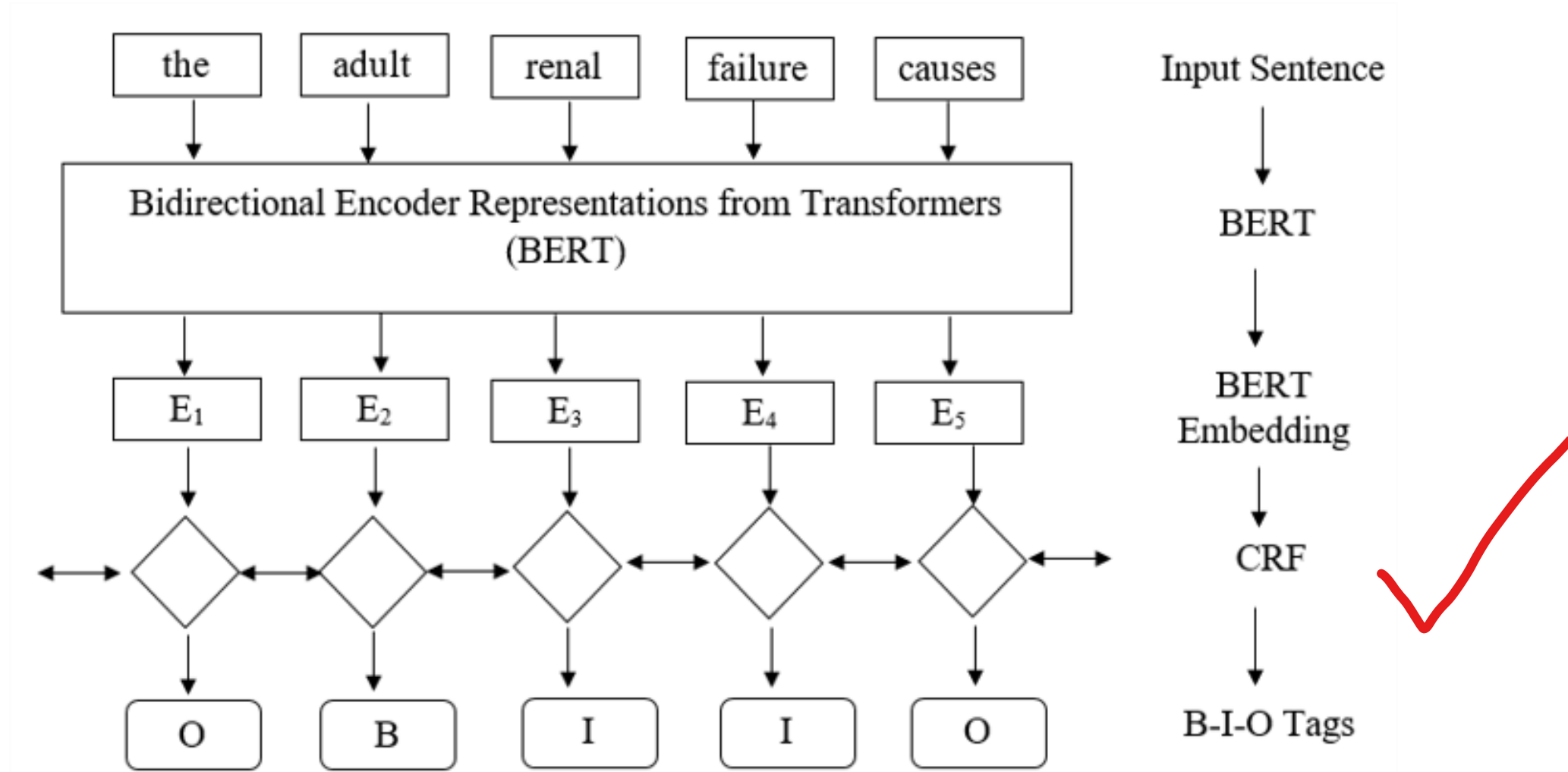
- Proposed model integrates Bidirectional Encoder Representations from Transformers (BERT) embedding with Conditional Random Fields (CRF).



Disease Named Entity Recognition (DNER)

Model

CRF = Conditional Random Field



Disease Named Entity Recognition (DNER)

CRF = Conditional Random Field

Model

