Transformer-based Language Models

Prachya Boonkwan (Arm)

Language and Semantic Technology Lab
National Electronics and Computer Technology Center, Thailand

prachya.boonkwan@nectec.or.th, kaamanita@gmail.com

URL ⇒ https://tinyurl.com/p8ezwzvm

Who? Me?

- Nickname: Arm (P'/N'/E' Arm, etc.)
- Born: Aug 1981
- Work: researcher at NECTEC since 2005
- Education
  - Alma mater: Triam Udom Suksa School
  - B.Eng & M.Eng, CPE Kasetsart University
  - Obtained OCSC Scholarship in early 2008
  - Did a PhD in Informatics (Computational Linguistics) at University of Edinburgh during 2008-2013 (4.5 years)

Princes Garden, Edinburgh (2010)

Who? Me?

- Nickname: Arm (P'/N'/E' Arm, etc.)
- Born: Aug 1981
- Work: researcher at NECTEC since 2005
- Honorable mentions
  - Developed Thai-English machine translation system for US Army's Cobra Gold Practice in 2006
  - Two best paper awards (as first author)
  - Gave a keynote speech in an academic conference
  - NSTDA's representative as future leader at *Science and Technology in Society* Forum 2018

Princes Garden, Edinburgh

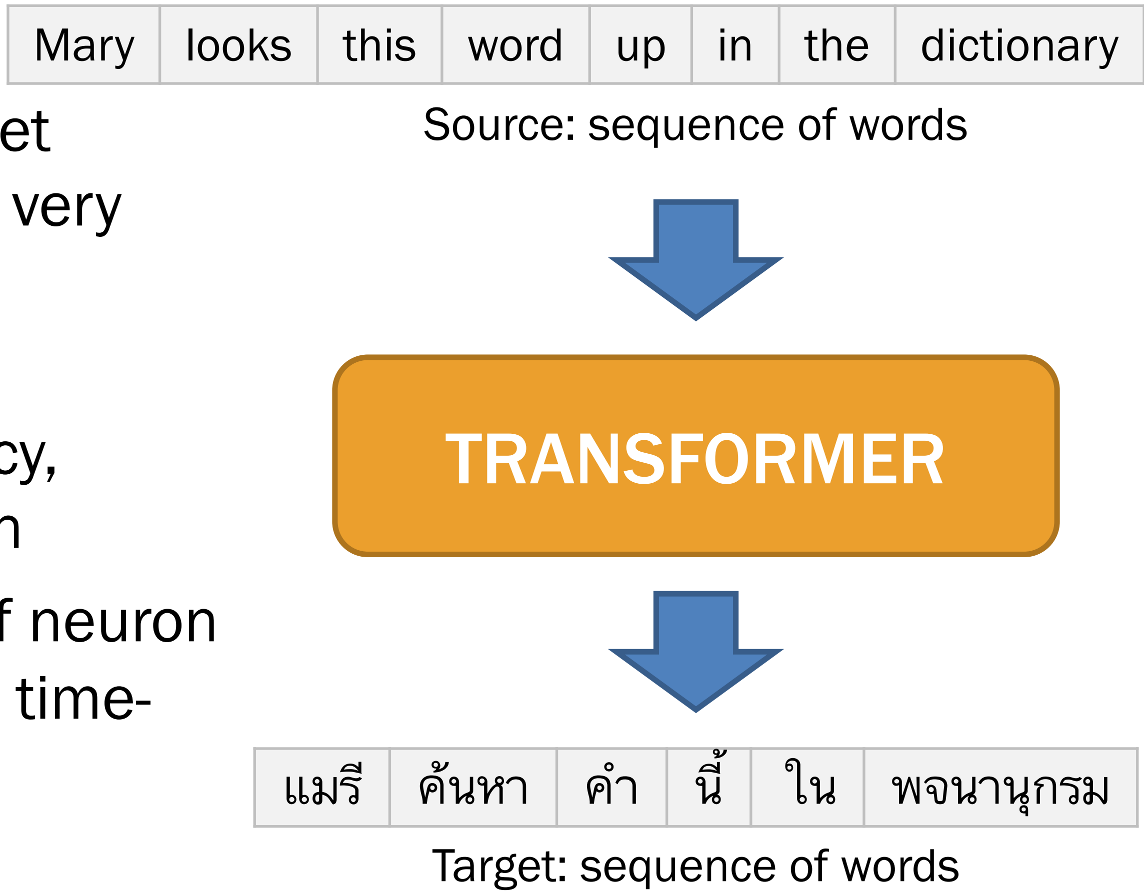
Outline

- Overview of the Transformer model
- Model interpretation
- Theoretical upper bounds
- BERT
- BERT variants
- Conclusion and discussion time

1. Overview of the Transformer Model

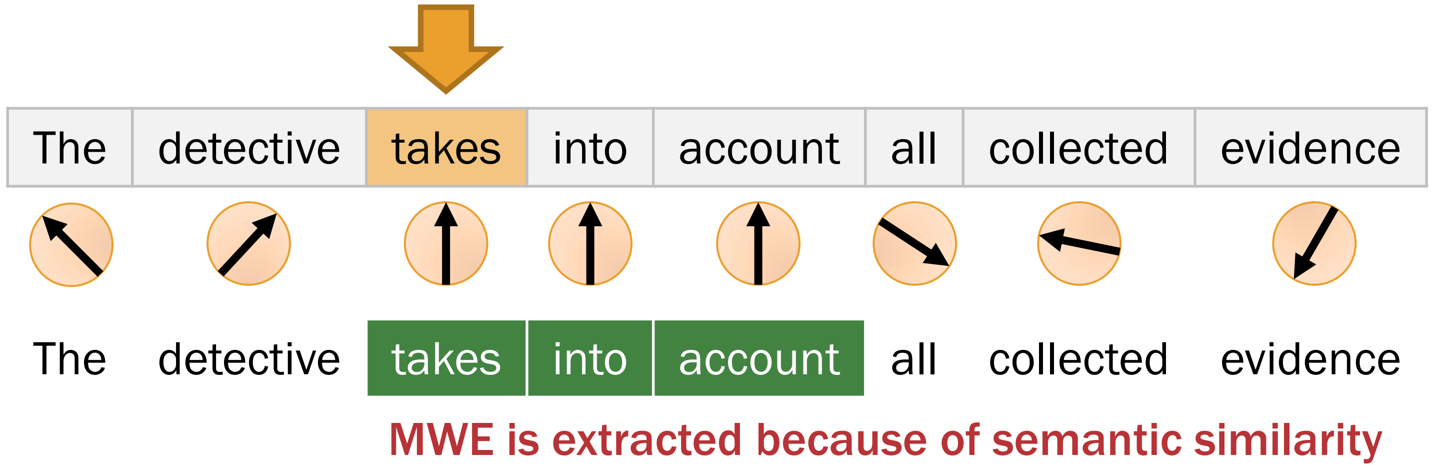
The Transformer (Vaswani et al., 2016)

- Sequence-to-sequence model
  - **Translation:** It learns how to produce a target sequence from a source sequence, given a very large dataset of sequence pairs
  - **Pros:** It is capable of learning multiword expressions, moderate-distance dependency, moderate reordering, and conceptualization
  - **Cons:** It consists of an expansive amount of neuron cells, and the training process can be quite time-consuming



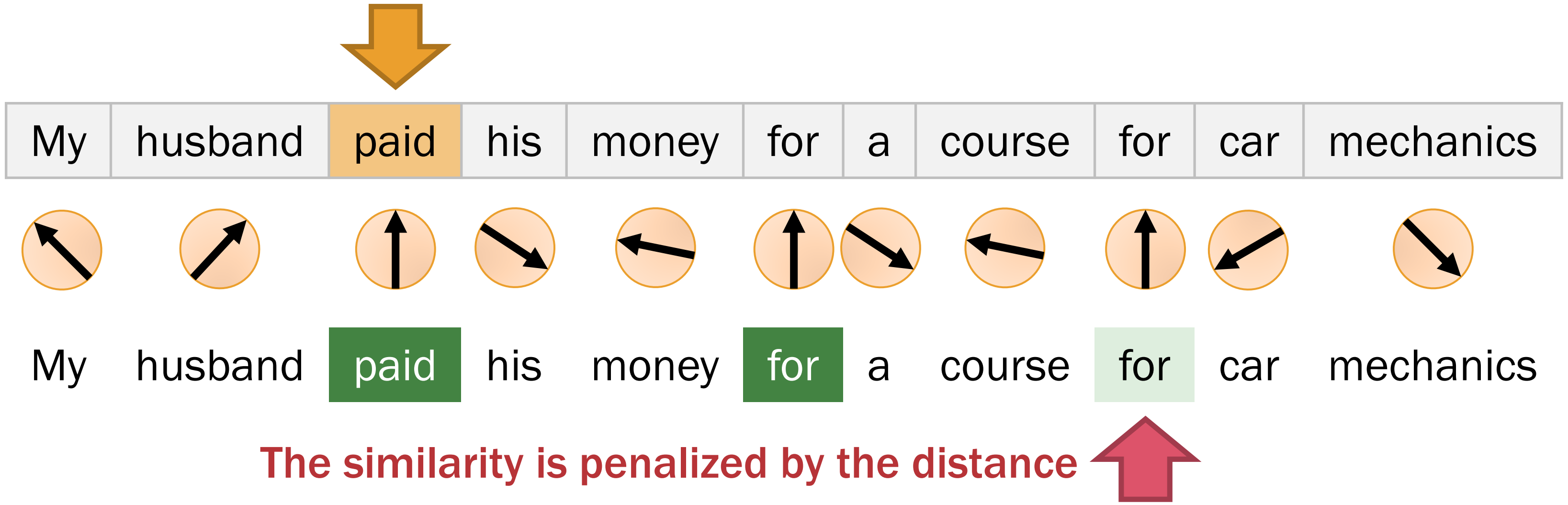
Pros: Multiword Expression (MWE)

- It recognizes the idiosyncratic collocations of at least 2 words
  - E.g. ‘peanut butter’, ‘car park’, ‘kick the bucket’, ‘take into account’, ‘break up’
  - It learns MWEs by comparing each word with the remaining to reveal semantic similarity



Pros: Moderate-Distance Dependency

- It recognizes word collocation that is separate within a moderate distance
  - E.g. 'look ___ up', 'ask ___ out', 'pay ___ for'
  - It learns moderate-distance dependency with semantic similarity and distance penalty



Pros: Moderate Reordering

- It learns to reorder words with next-word prediction (language model), cross-lingual semantic similarity, and distance penalty

Next-word prediction takes into account an <u>entire</u> input sequence

your	stable	financial	status	PREDICTION	
your	stable	financial	**status**	∅ ⇒ สถานะ	สถานะ
your	stable	**financial**	status	สถานะ ⇒ ทาง	สถานะ ทาง
your	stable	**financial**	status	ทาง ⇒ การเงิน	สถานะ ทาง การเงิน
your	**stable**	financial	status	การเงิน ⇒ ที่	สถานะ ทาง การเงิน ที่
your	**stable**	financial	status	ที่ ⇒ มั่นคง	สถานะ ทาง การเงิน ที่ มั่นคง
your	stable	financial	status	มั่นคง ⇒ ของ	สถานะ ทาง การเงิน ที่ มั่นคง ของ
your	stable	financial	status	ของ ⇒ ท่าน	สถานะ ทาง การเงิน ที่ มั่นคง ของ ท่าน

Pros: Conceptualization

- It learns to conceptualize a long subsequence into a shorter one with semantic similarity and distance penalty
  - E.g. ‘initiated a scheme for building ___’ is conceptualized into ‘invented’ and consequently translated into ‘ประดิษฐ์’

									PREDICTION					
Stevenson	initiated	a	scheme	for	building	the	first	locomotive						
Stevenson	initiated	a	scheme	for	building	the	first	locomotive	∅ ⇒ สตีเวนสัน	สตีเวนสัน				
Stevenson	initiated	a	scheme	for	building	the	first	locomotive	สตีเวนสัน ⇒ ประดิษฐ์	สตีเวนสัน	ประดิษฐ์			
Stevenson	initiated	a	scheme	for	building	the	first	locomotive	ประดิษฐ์ ⇒ รถจักรไอน้ำ	สตีเวนสัน	ประดิษฐ์	รถจักรไอน้ำ		
Stevenson	initiated	a	scheme	for	building	the	first	locomotive	รถจักรไอน้ำ ⇒ คัน	สตีเวนสัน	ประดิษฐ์	รถจักรไอน้ำ	คัน	
Stevenson	initiated	a	scheme	for	building	the	first	locomotive	คัน ⇒ แรก	สตีเวนสัน	ประดิษฐ์	รถจักรไอน้ำ	คัน	แรก

Notable Applications in NLP

Applications	Descriptions	Input	Output	What is Learned?
Neural machine translation	Convert a text from the source language to the target language	Word sequence in the source language	Word sequence in the target language	• Word alignment (cross-lingual semantic similarity) • MWEs in both languages (semantic similarity)
Abstractive summarization	Translate a text into a shorter version in the same language	Word sequence of full text	Word sequence of summary	• MWEs in the language • Pronoun substitution • Conceptualization
Image captioning	Explain an image with a short description	Sequence of image fragments	Word sequence of image caption	• Image-to-word alignment (multimodal semantic similarity) • MWEs in the language
Speech recognition	Transcribe a sequence of audio signal into phonetic representation (IPA)	Sequence of audio signals (frequency domain)	Sequence of phonetic representation	• Sound-to-transcription alignment (multimodal semantic similarity) • Phonetic processes in the language

‘Plausible’ Applications in NLP

Applications	Descriptions	Input	Output	What is Learned?
Sequential tagging	Annotate each token of a given sequence with a linguistic tag (e.g. POS and NE)	Sequence of characters or words	Sequence of words with linguistic tags	• Token-to-tag alignment • Contextual clues for linguistic annotation • Joint annotation model
Syntactic parsing	Annotate a sequence of words with a syntactic structure	Sequence of words	Sequence of parsing actions (shift, reduce, accept, backtrack)	• Word-to-tree alignment • Parsing model based on semantic similarity
Word segmentation with term normalization	Tokenize a given string into a word sequence and normalize non-canonical terms	Sequence of characters	Sequence of words	• MWEs in the language • Spelling rules
Relation extraction	Determine the relationship between the main verb and its arguments	Sequence of words	Knowledge graph	• Verb-to-argument relationship based on semantic similarity • MWEs in the language

2. Model Interpretation

Scaled Dot-Product Attention

- Semantic similarity ⇒ search engine
  - Query is compared against each key with dot product
  - The more similar the key is to the query, the more weight its value will get

Simple Form

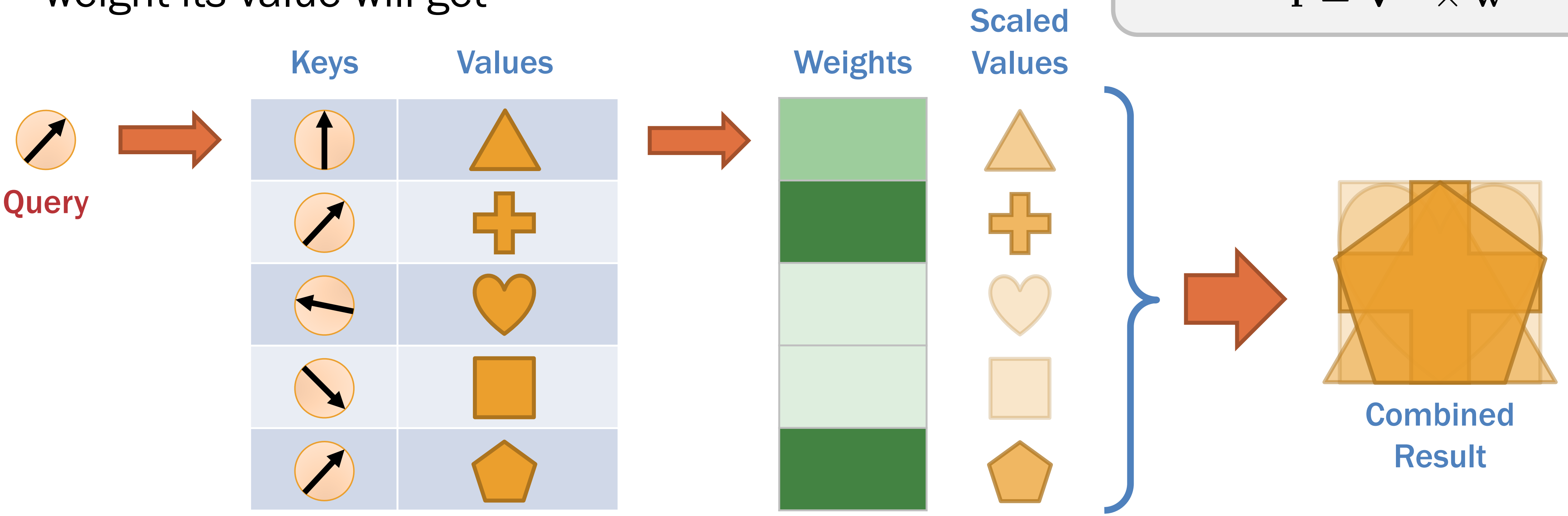
$$w_i \propto \mathbf{k}_i \cdot \mathbf{q}$$

$$\mathbf{r} = \sum_{i=1}^{N} w_i \mathbf{v}_i$$

Matrix Form

$$\mathbf{w} = \mathrm{Softmax}(\mathbf{K} \times \mathbf{q})$$

$$\mathbf{r} = \mathbf{V}^\top \times \mathbf{w}$$



Scaled Dot-Product Attention

- Semantic similarity ⇒ search engine
  - Query is compared against each key with dot product
  - The more similar the key is to the query, the more weight its value will get

Simple Form

$$w_i \propto \mathbf{k}_i \cdot \mathbf{q}$$

$$\mathbf{r} = \sum_{i=1}^{N} w_i \mathbf{v}_i$$

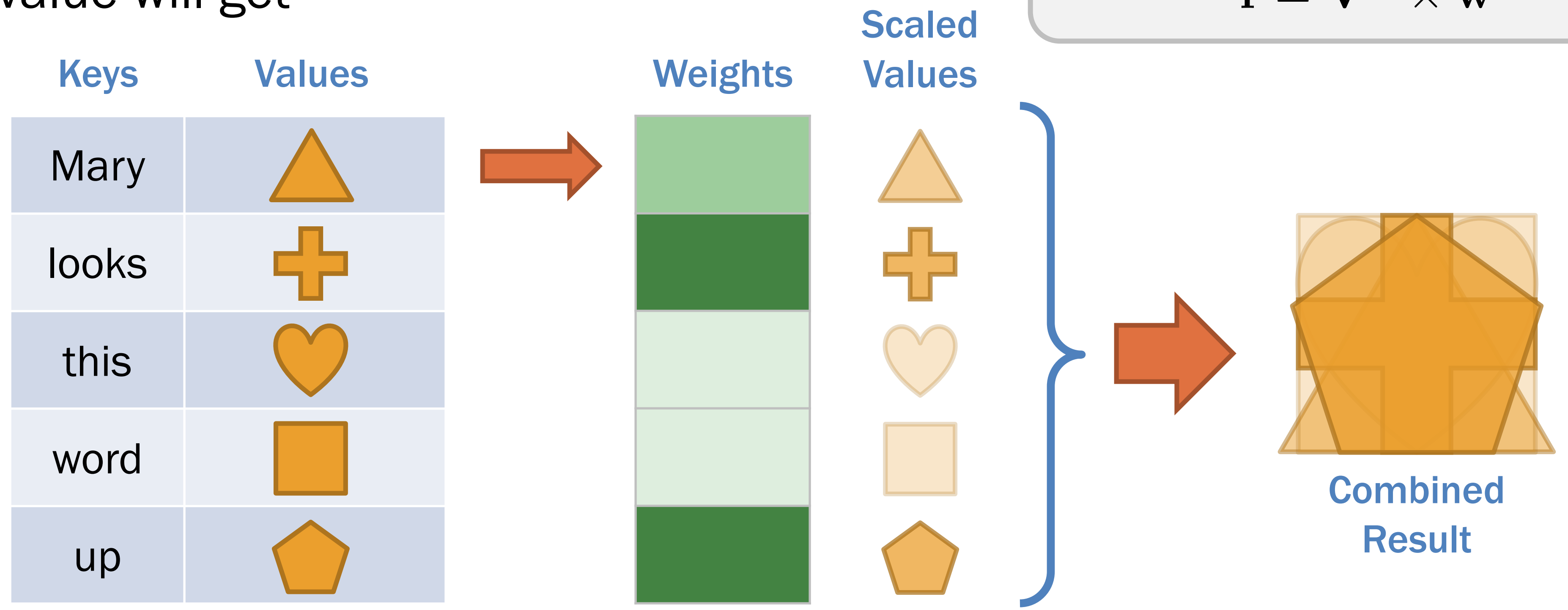
Matrix Form

$$\mathbf{w} = \mathrm{Softmax}(\mathbf{K} \times \mathbf{q})$$

$$\mathbf{r} = \mathbf{V}^\top \times \mathbf{w}$$

'looks'

Query



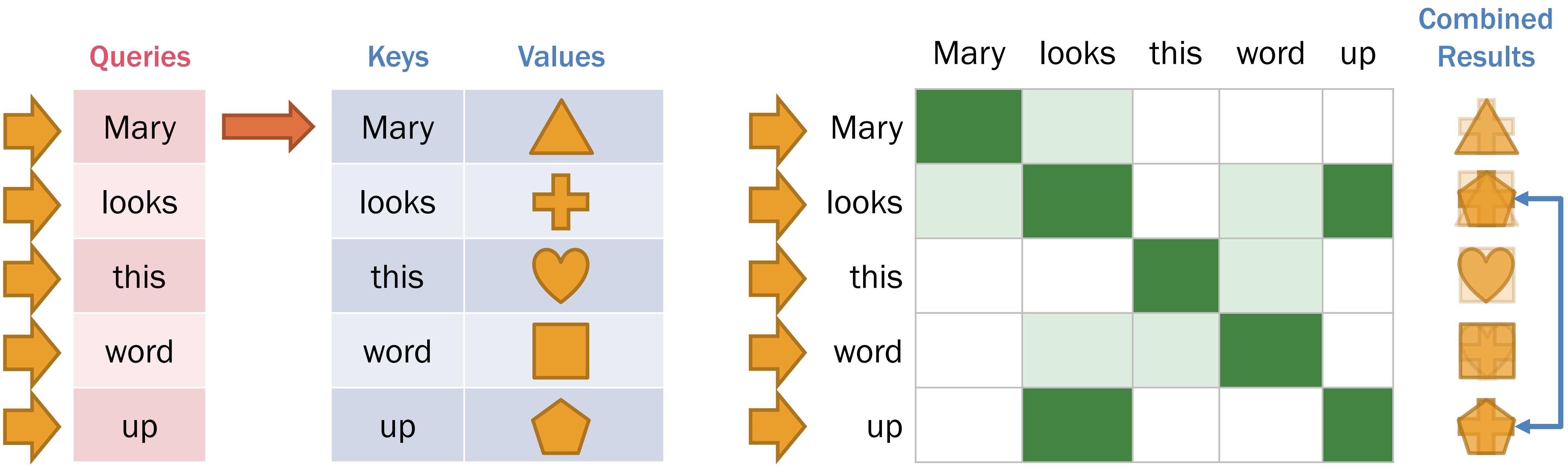
For word sequence, collocating words are semantically similar to each other e.g. 'looks ___ up'

Self-Attention

- Scaled dot-product attention whose queries and keys are the same
- Collocations will have almost similar results

Matrix Form

$$\mathbf{W} = \mathrm{Softmax}(\mathbf{K} \times \mathbf{K}^\top)$$
$$\mathbf{R} = \mathbf{W} \times \mathbf{V}$$

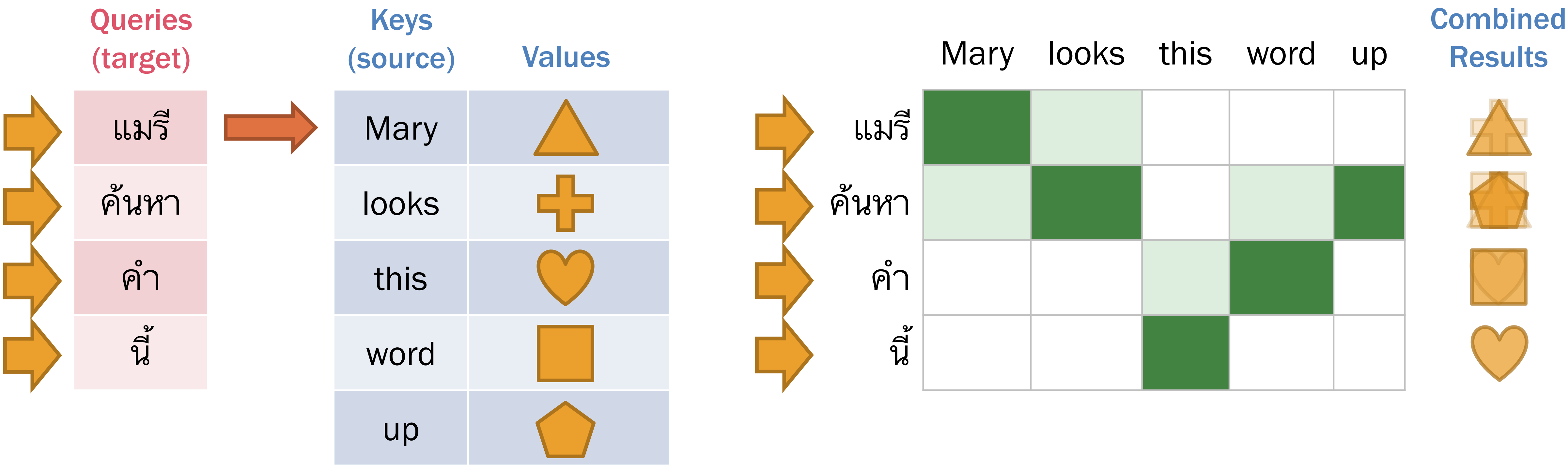


Alignment Attention

- Scaled dot-product attention whose queries are the target and whose keys are the source
- Collocation alignment via semantic similarity

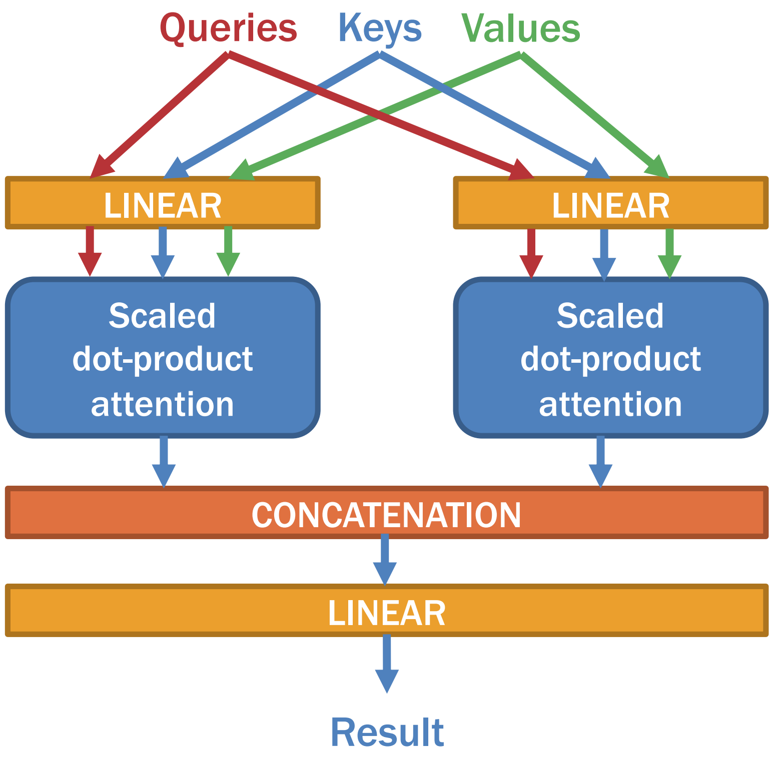
Matrix Form

$$\mathbf{W} = \mathrm{Softmax}(\mathbf{Q} \times \mathbf{K}^{\top})$$
$$\mathbf{R} = \mathbf{W} \times \mathbf{V}$$

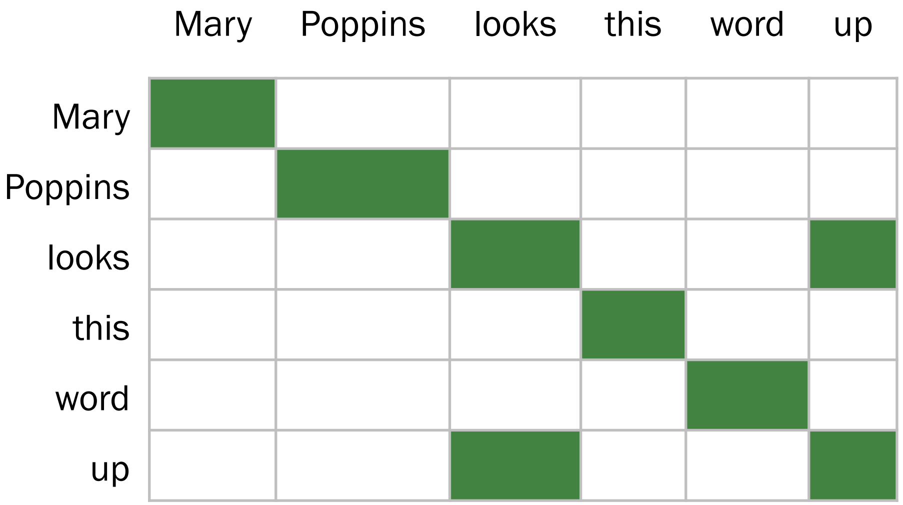


Multihead Attention

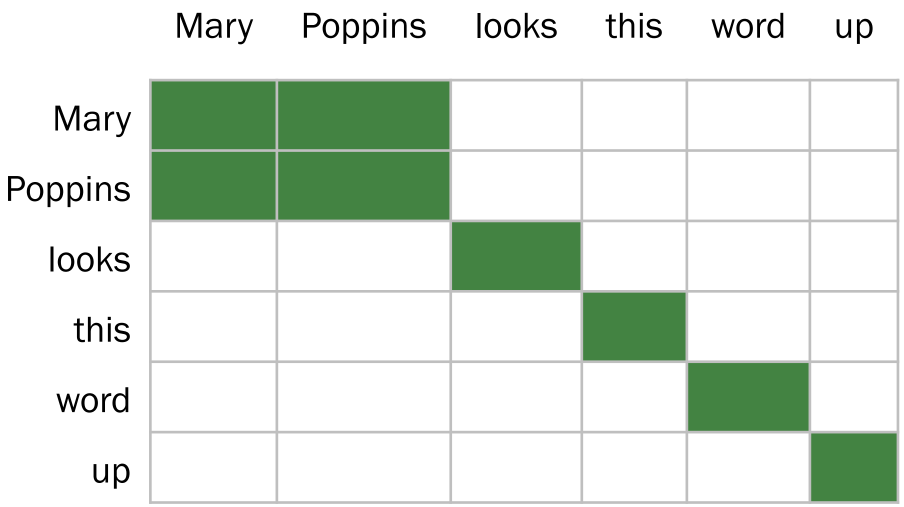
- Scaled dot-product attention has a drawback
  - It recognizes **only one** type of word collocation
  - If we assume more than one type of word collocation per sequence, then we have to combine multiple attention heads [default = 8 heads]

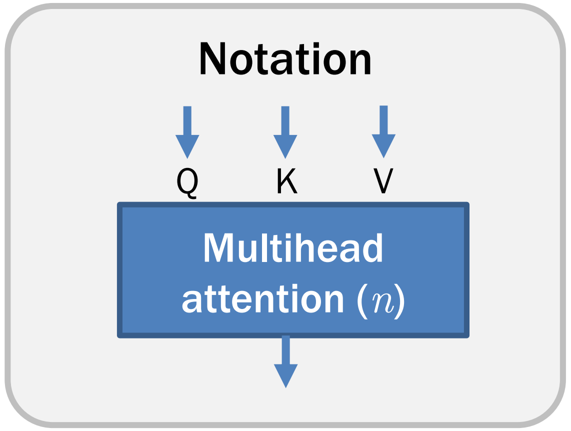


HEAD 1 (looks ___ up)



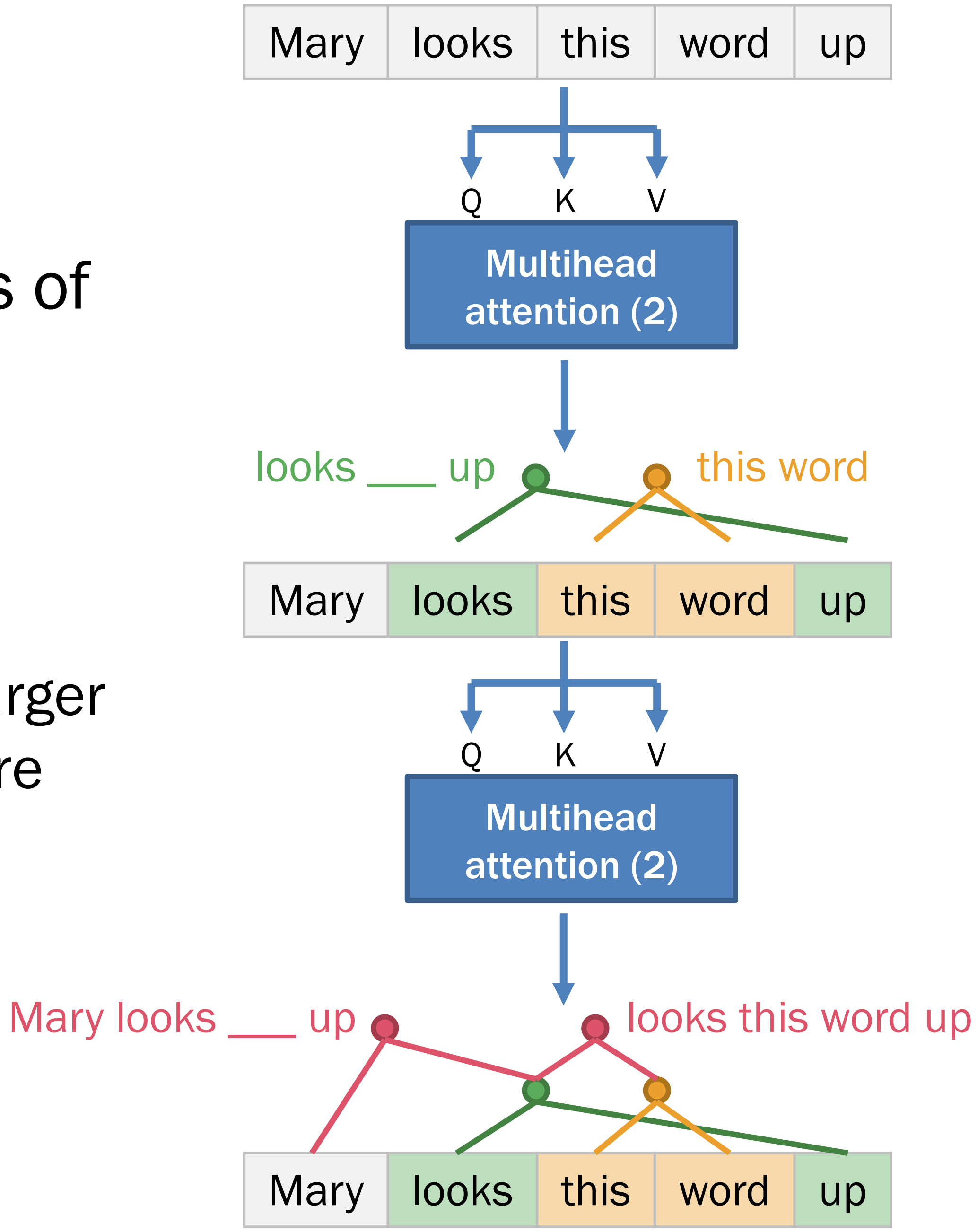
HEAD 2 (Mary Poppins)





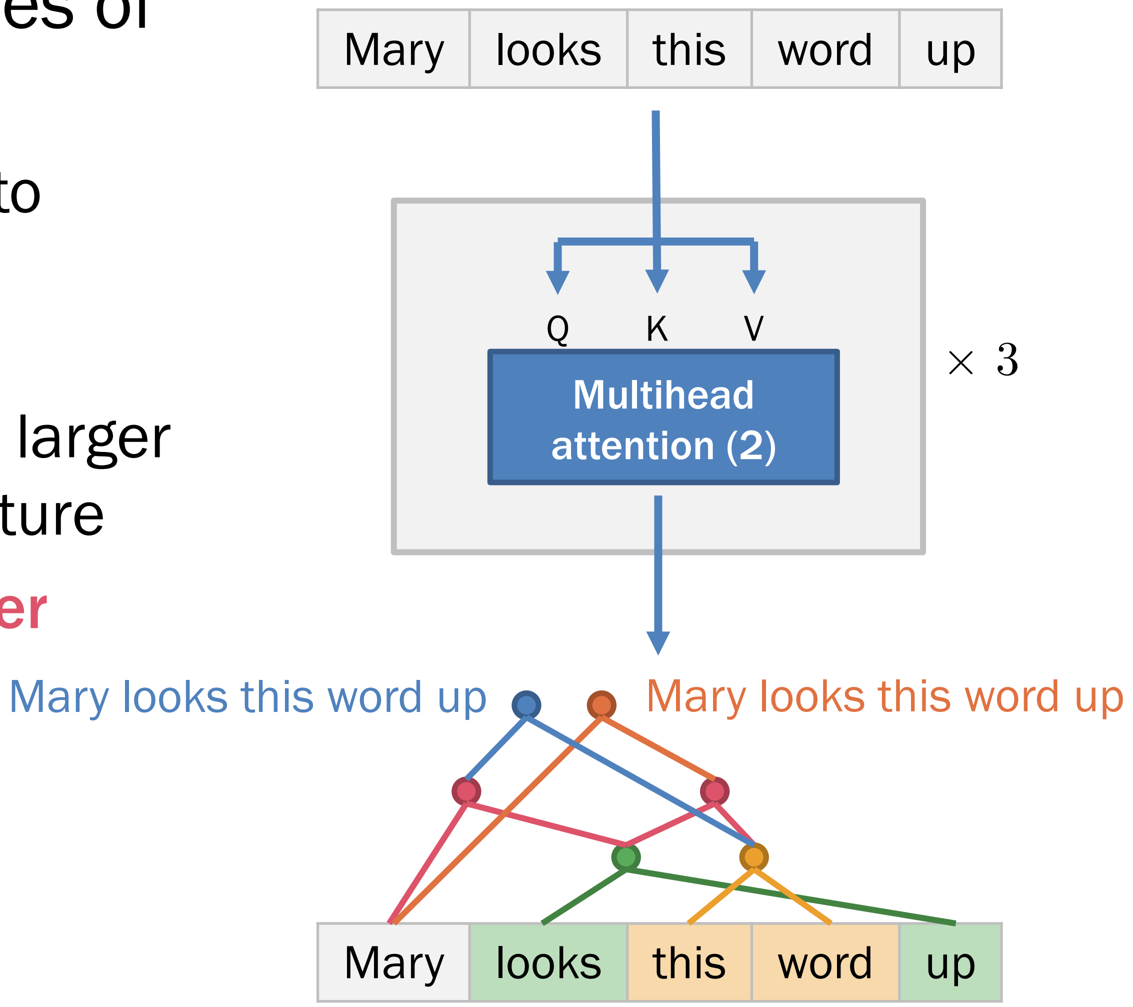
Phrase Structure

- H-head self-attention recognizes H types of word collocation per sequence
  - One layer can combine consecutive words to become a phrase
  - More layers of multihead self-attention can combine consecutive phrases to become a larger phrase or even a sentence $\Rightarrow$ phrase structure
  - Each layer is simply called an **encoding layer**



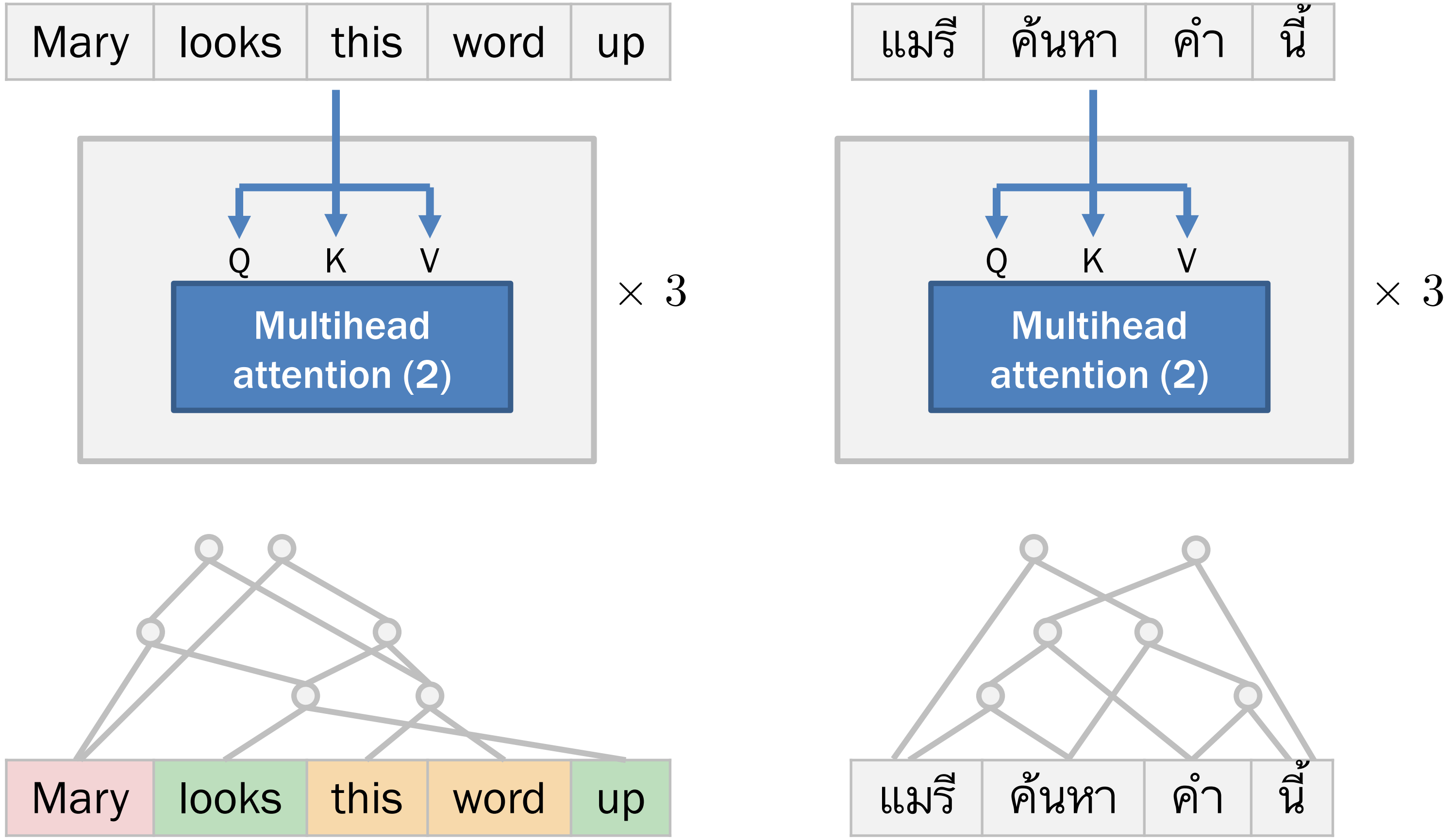
Phrase Structure

- H-head self-attention recognizes H types of word collocation per sequence
  - One layer can combine consecutive words to become a phrase
  - More layers of multihead self-attention can combine consecutive phrases to become a larger phrase or even a sentence $\Rightarrow$ phrase structure
  - Each layer is simply called an **encoding layer**



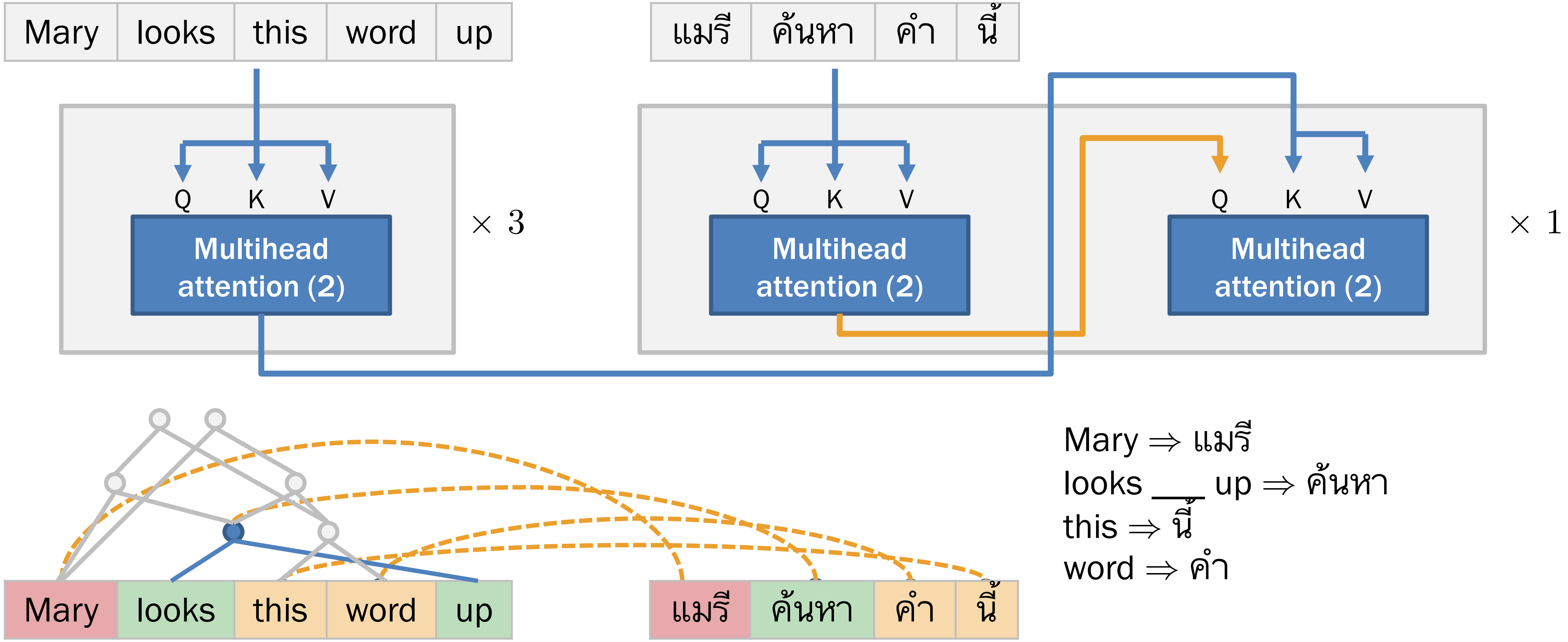
Alignment of Phrase Structures

- H-head alignment attention recognizes H pairs of phrase structures



Alignment of Phrase Structures

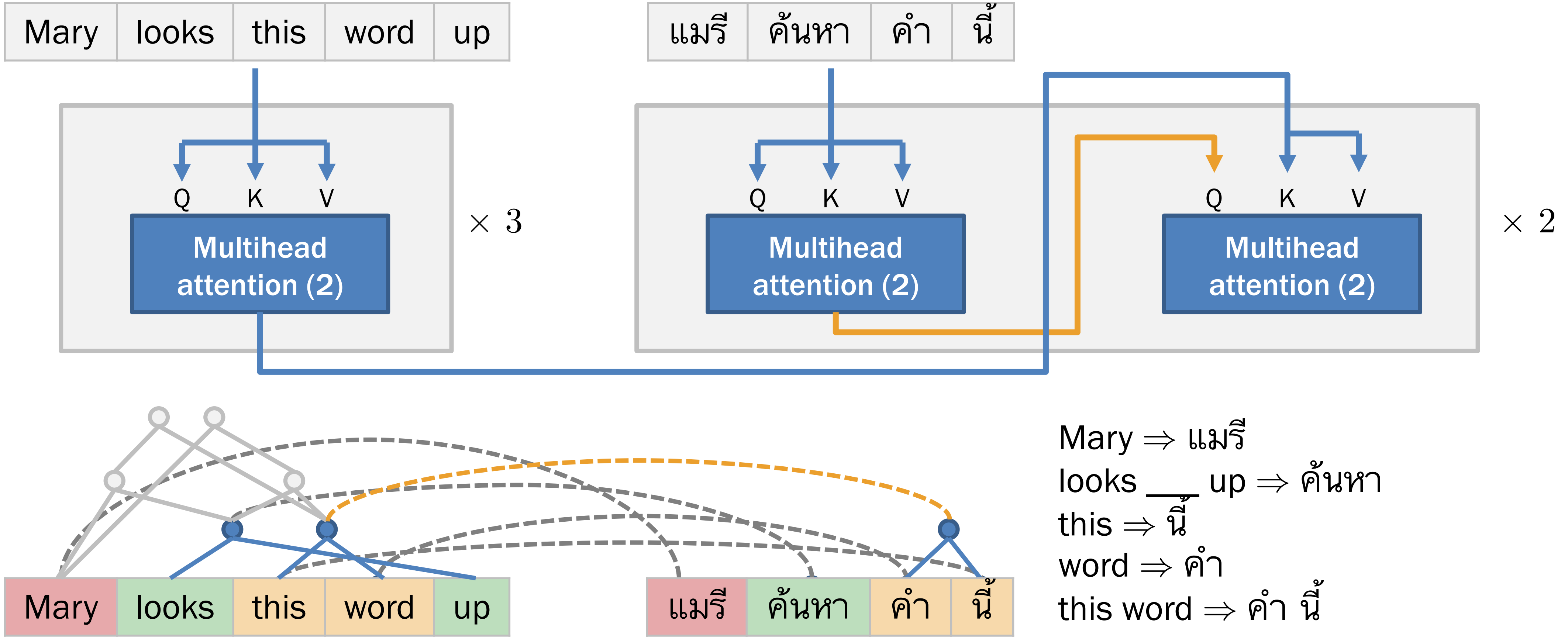
- H-head alignment attention recognizes H pairs of phrase structures $\Rightarrow$ **decoding layer**



Mary $\Rightarrow$ แมรี
looks ___ up $\Rightarrow$ ค้นหา
this $\Rightarrow$ นี้
word $\Rightarrow$ คำ

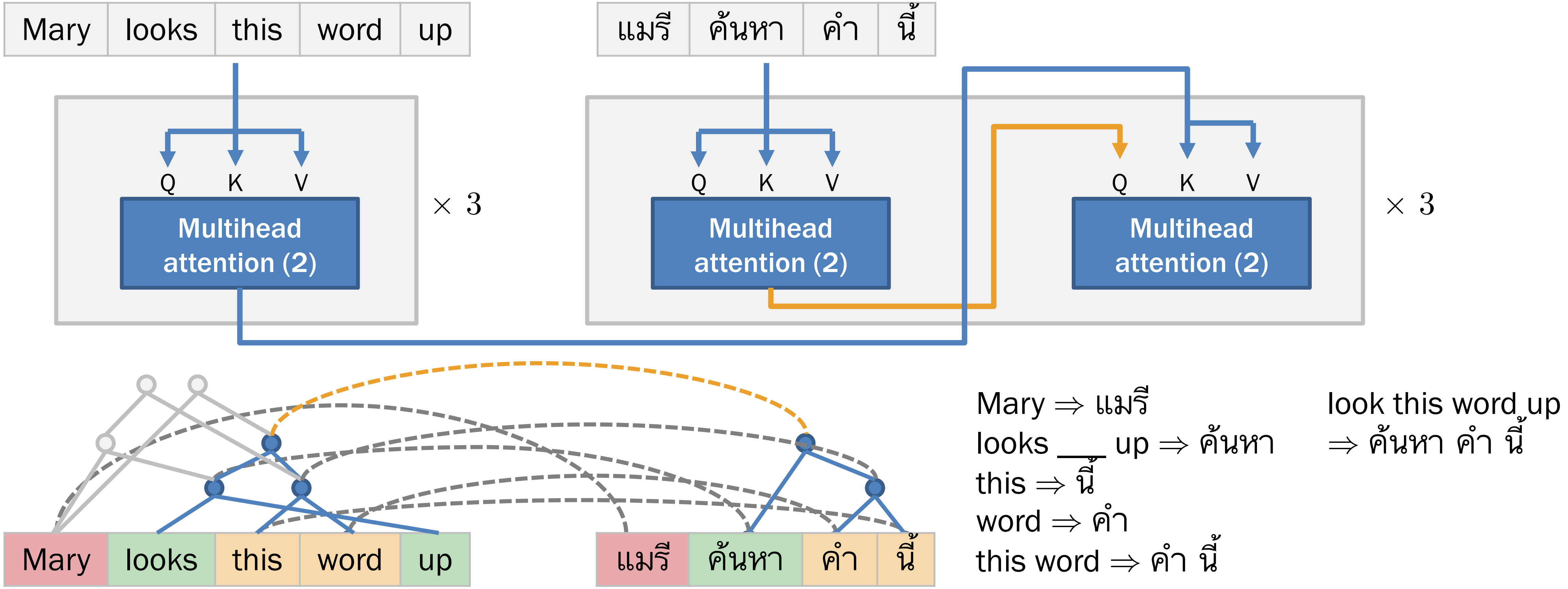
Alignment of Phrase Structures

- H-head alignment attention recognizes H pairs of phrase structures $\Rightarrow$ decoding layer



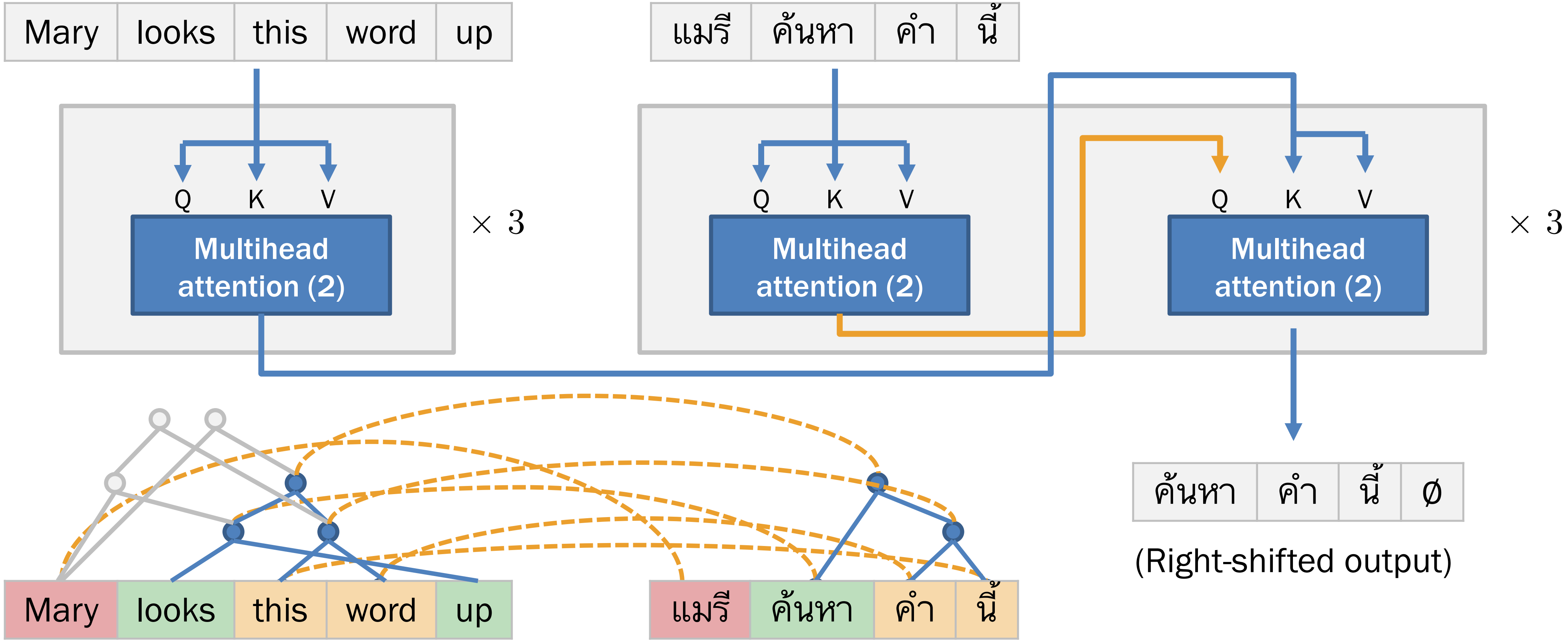
Alignment of Phrase Structures

- H-head alignment attention recognizes H pairs of phrase structures ⇒ decoding layer

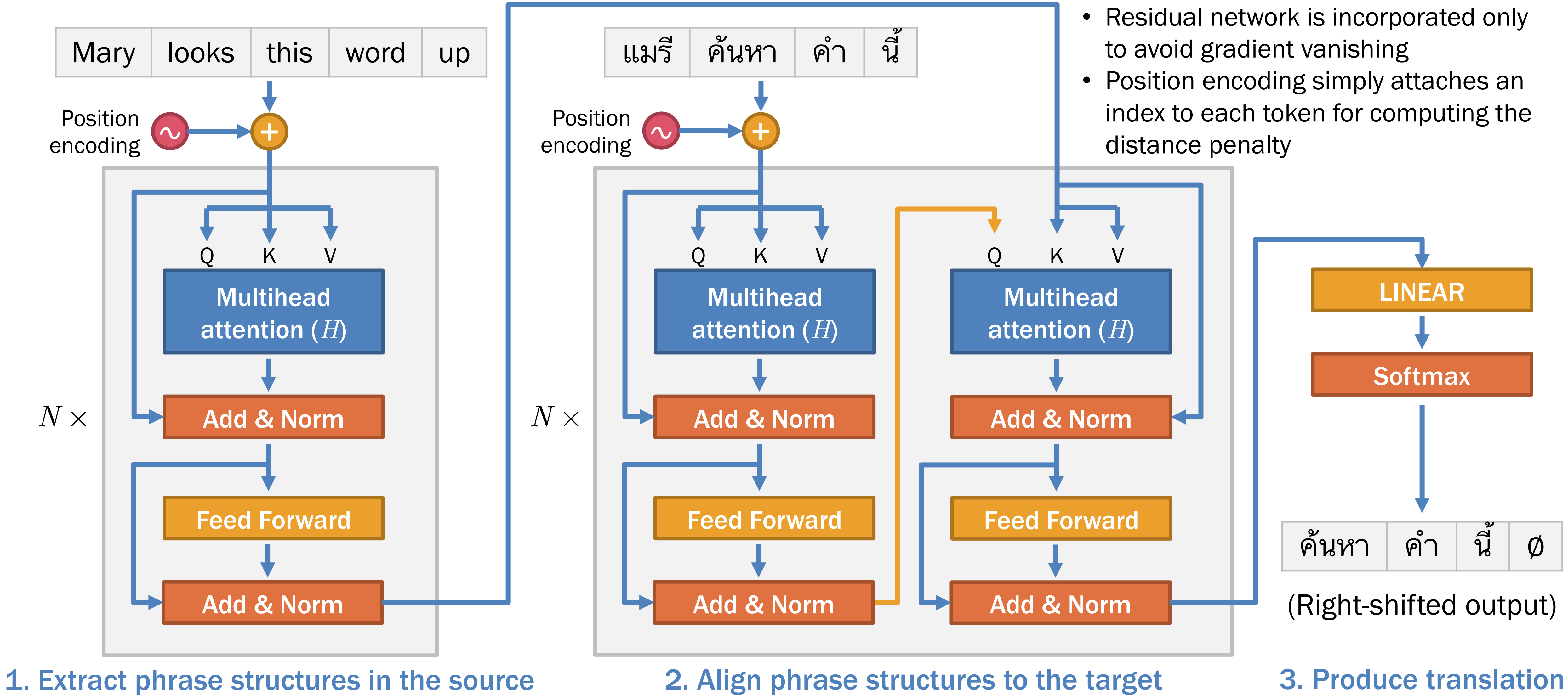


Alignment of Phrase Structures

- H-head alignment attention recognizes H pairs of phrase structures $\Rightarrow$ decoding layer

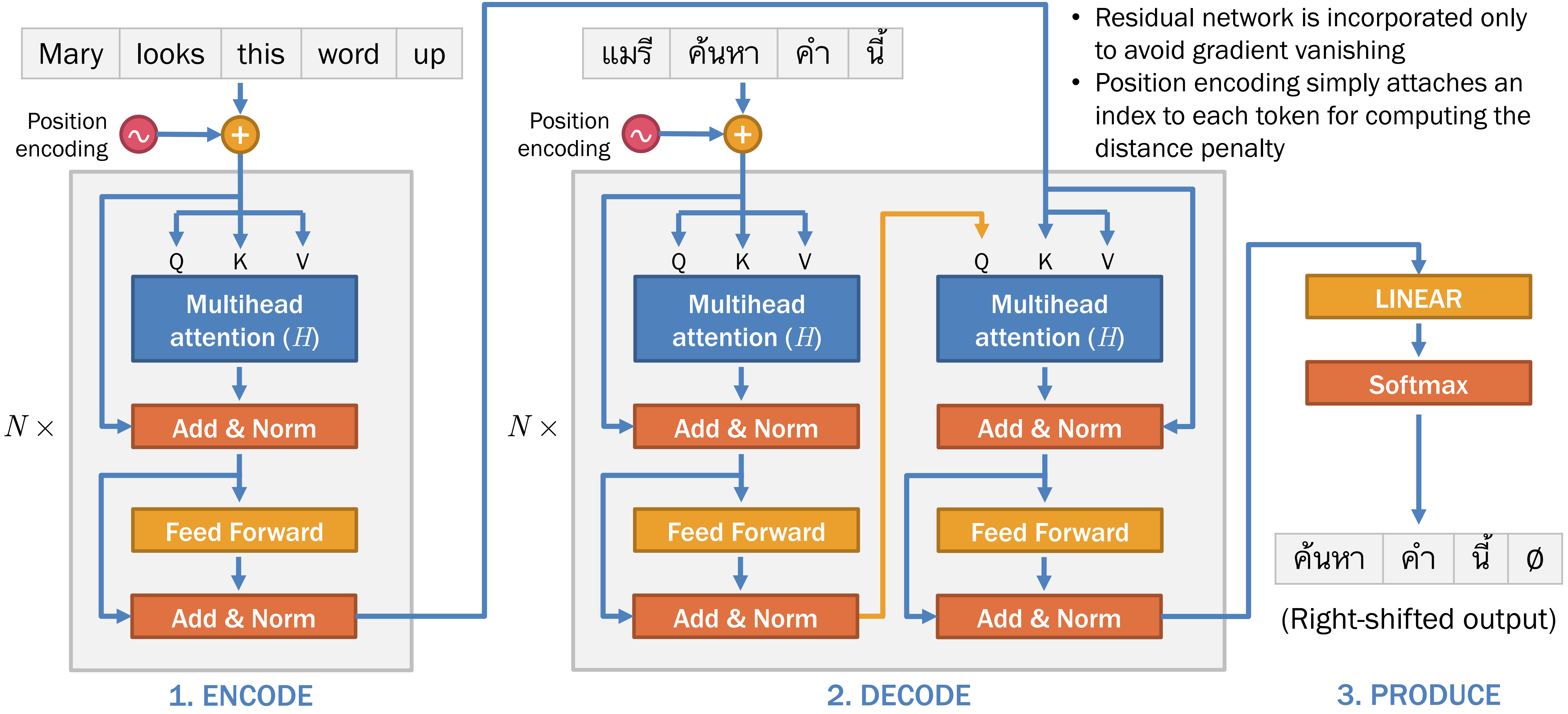


Overview of the Transformer Model



- Residual network is incorporated only to avoid gradient vanishing
- Position encoding simply attaches an index to each token for computing the distance penalty

Overview of the Transformer Model



- Residual network is incorporated only to avoid gradient vanishing
- Position encoding simply attaches an index to each token for computing the distance penalty

Evaluation: BLEU Score

- BiLingual Evaluation Understudy (BLEU)
- n-gram precision = ratio between the matched n-grams against the candidate

$$\text{1-gram prec} = \frac{7}{10}$$

$$\text{2-gram prec} = \frac{4}{9}$$

$$\text{3-gram prec} = \frac{1}{8}$$

Reference: the Iraqi weapons are to be handed over to the army within two weeks

Candidate (decoded): in two weeks Iraqi weapons will give to the army

$$\text{BLEU} = \left(\prod_{n=1}^{3} p_n \right)^{1/3} = \left(\frac{7}{10} \times \frac{4}{9} \times \frac{1}{8} \right)^{1/3}$$

Evaluation: ROUGE Scores

- ROUGE-n = ratio between matched n-grams **against the reference**
- ROUGE-L = geo. mean of ratios between the longest common subsequence **and both texts**

$$\text{ROUGE-1} = \frac{7}{14}$$

$$\text{ROUGE-2} = \frac{4}{13}$$

$$\text{ROUGE-3} = \frac{1}{12}$$

Reference: the Iraqi weapons are to be handed over **to the army** within **two weeks**

Candidate (decoded): in **two weeks** Iraqi weapons will give **to the army**

$$\text{Prec} = \frac{5}{10}$$

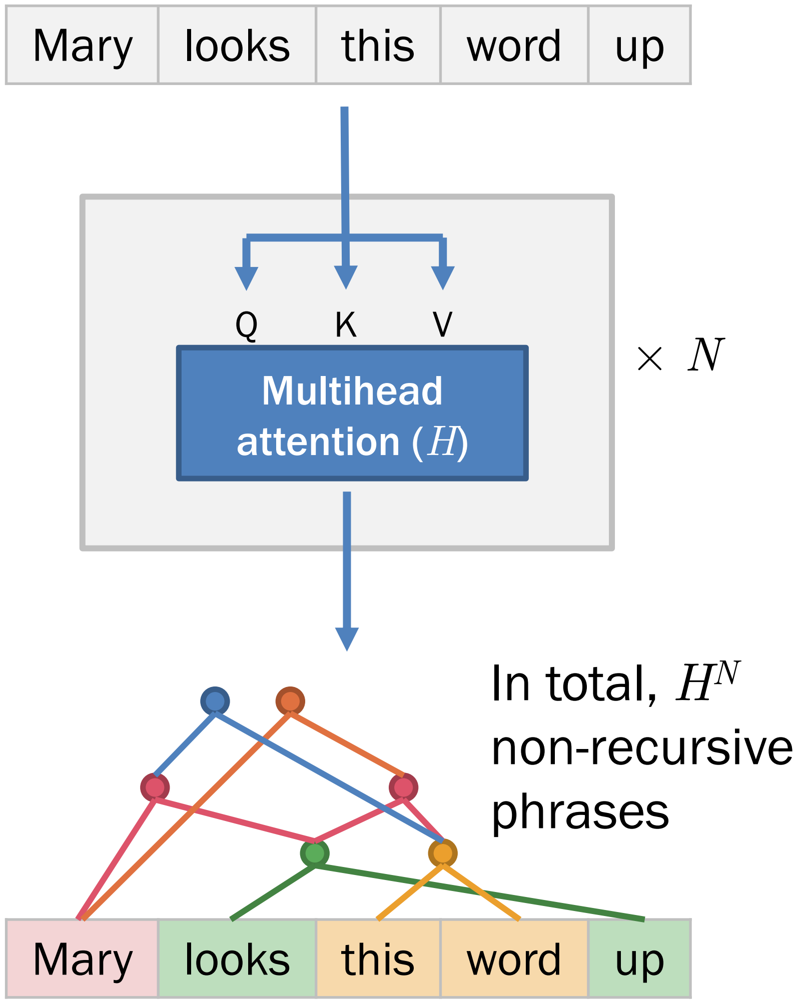
$$\text{Rec} = \frac{5}{14}$$

$$\text{ROUGE-}L = \frac{2}{\frac{1}{\text{Prec}} + \frac{1}{\text{Rec}}}$$

3. Theoretical Upper Bounds

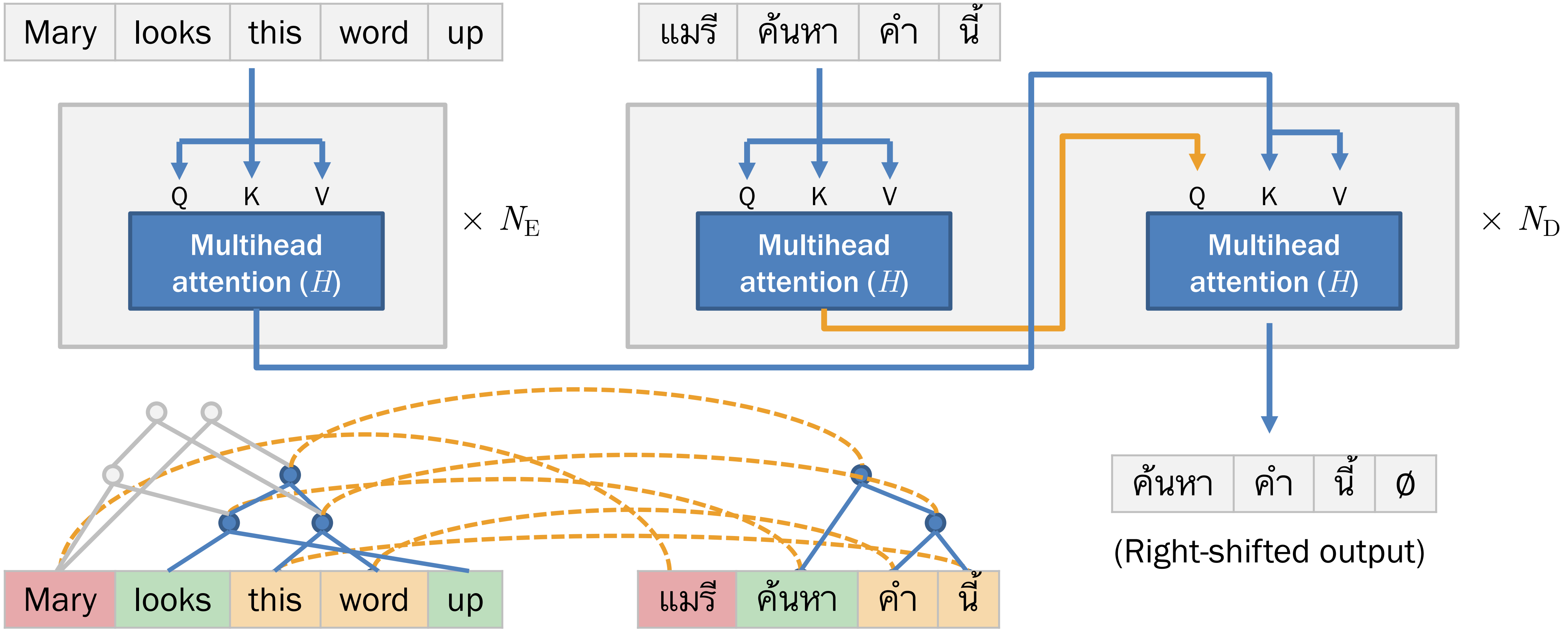
Encoding Phrase Structures

- **Limitation:** One self-attention head learns only one type of word collocation
  - H-head self-attention learns at best H types of word collocation
  - Adding one self-attention head on top of H-head self-attention helps learn a phrase structure of these H types of word collocation
  - So, adding H-head self-attention to H-head self-attention helps learn H^2 possible phrases
  - Therefore, N layers of H-head self-attention learns H^N possible non-recursive phrases
  - **Default:** $H=8$, $N=6 \Rightarrow$ 262,144 possible phrases



Decoding Phrase Structures

- **Limitation:** Encoder-decoder learns at best $H^{N_\mathrm{E}+N_\mathrm{D}}$ non-recursive translation pairs
- **Default:** H=8, N_E=6, N_D=6 $\Rightarrow$ 6.87 billion possible pairs



Effects of Upper Bound Violation

- If there are $> H^N$ phrase structures
  - Distinct phrases may be encoded as the same values in the multihead self-attention
  - **Encoding:** It causes lexical mistranslation
  - **Decoding:** It causes under-generation and over-generation
- If there are $> H^{N_\mathrm{E}+N_\mathrm{D}}$ translation pairs
  - Distinct translation pairs may be stored as the same pairs in the alignment attention
  - This results in phrase mistranslation, under-generation, and over-generation

4. BERT

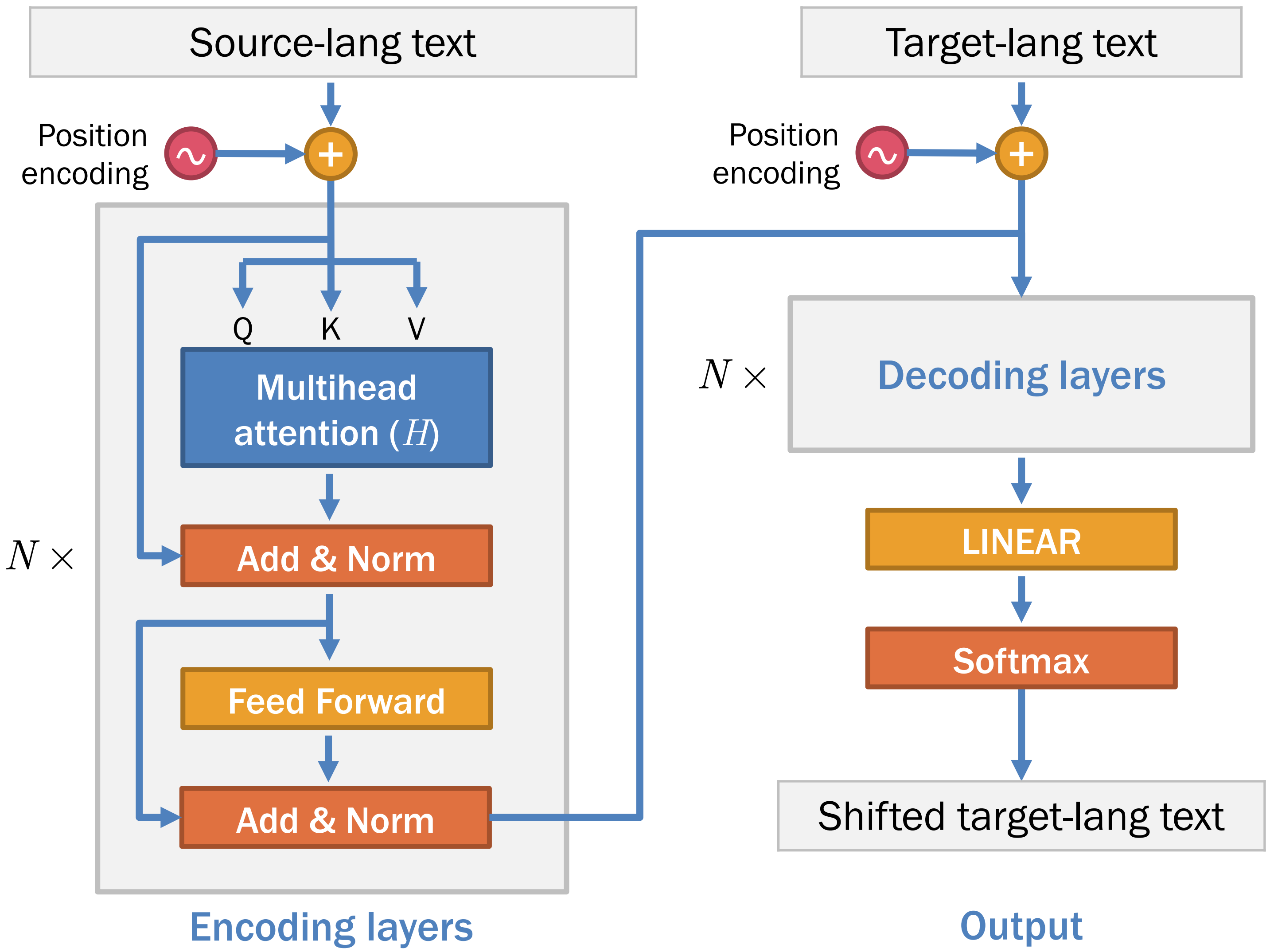
<u>B</u>idirectional <u>E</u>ncoder <u>R</u>epresentations from <u>T</u>ransformer

BERT (Devlin et al., 2018)

- Bidirectional Encoder Representations from Transformer
  - Pretrained Transformer model with multilayer bidirectional encoders
  - Contextual representations: vector repr of each word varies by position
  - Trained on BooksCorpus (800M words) + Wikipedia (2,500M words)

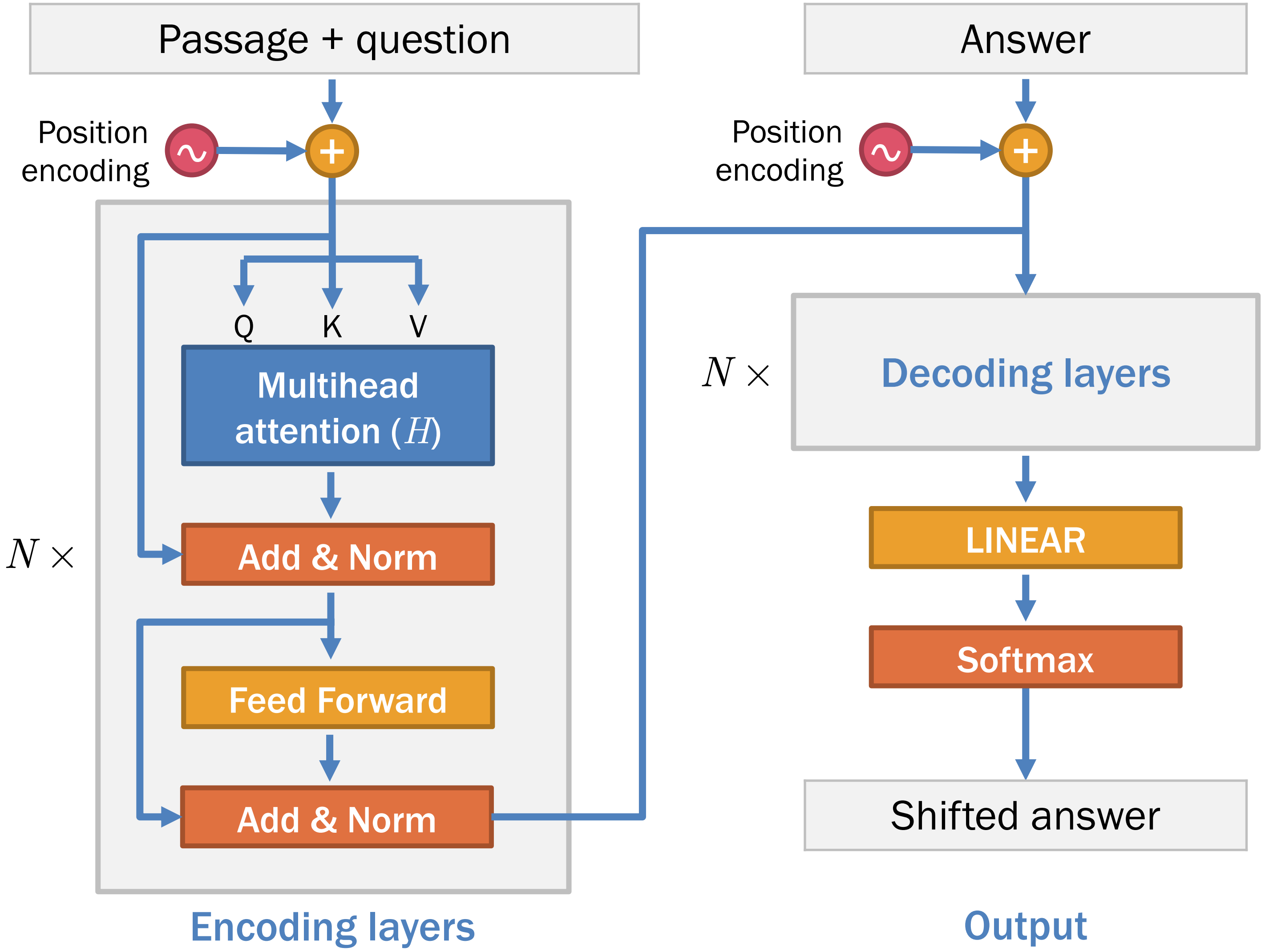
	BERT base	BERT large
Encoding layers	12	24
Attention heads	12	16
Hidden dimensions	768	1,024
Parameters	110M	340M

Training BERT out of the Transformer



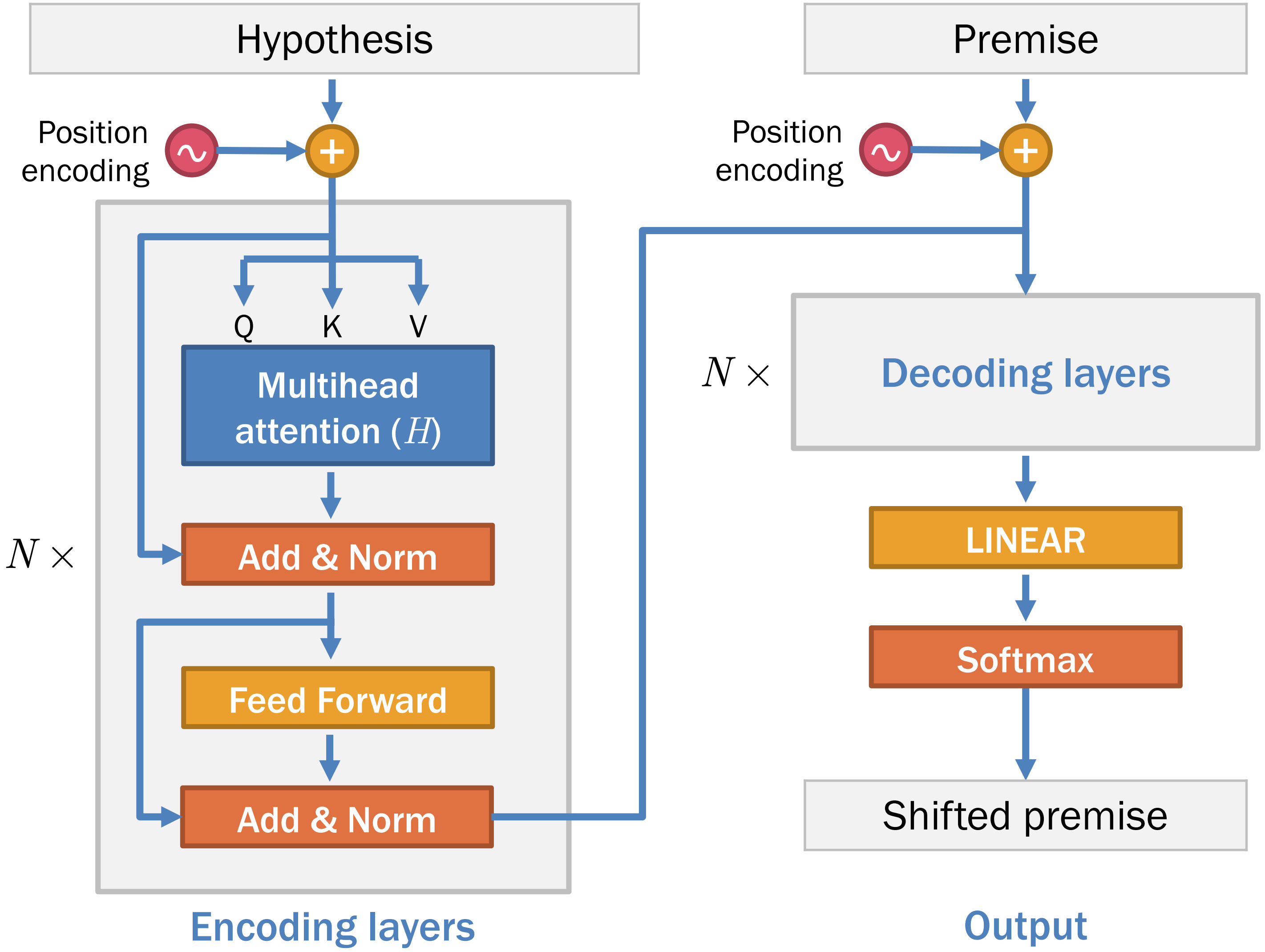
- BERT can be train via multiple downstream tasks
  - Machine translation
  - Question answering (SQUAD)
  - Inference in natural language (NLI in GLUE Dataset)
  - Abstractive summarization

Training BERT out of the Transformer



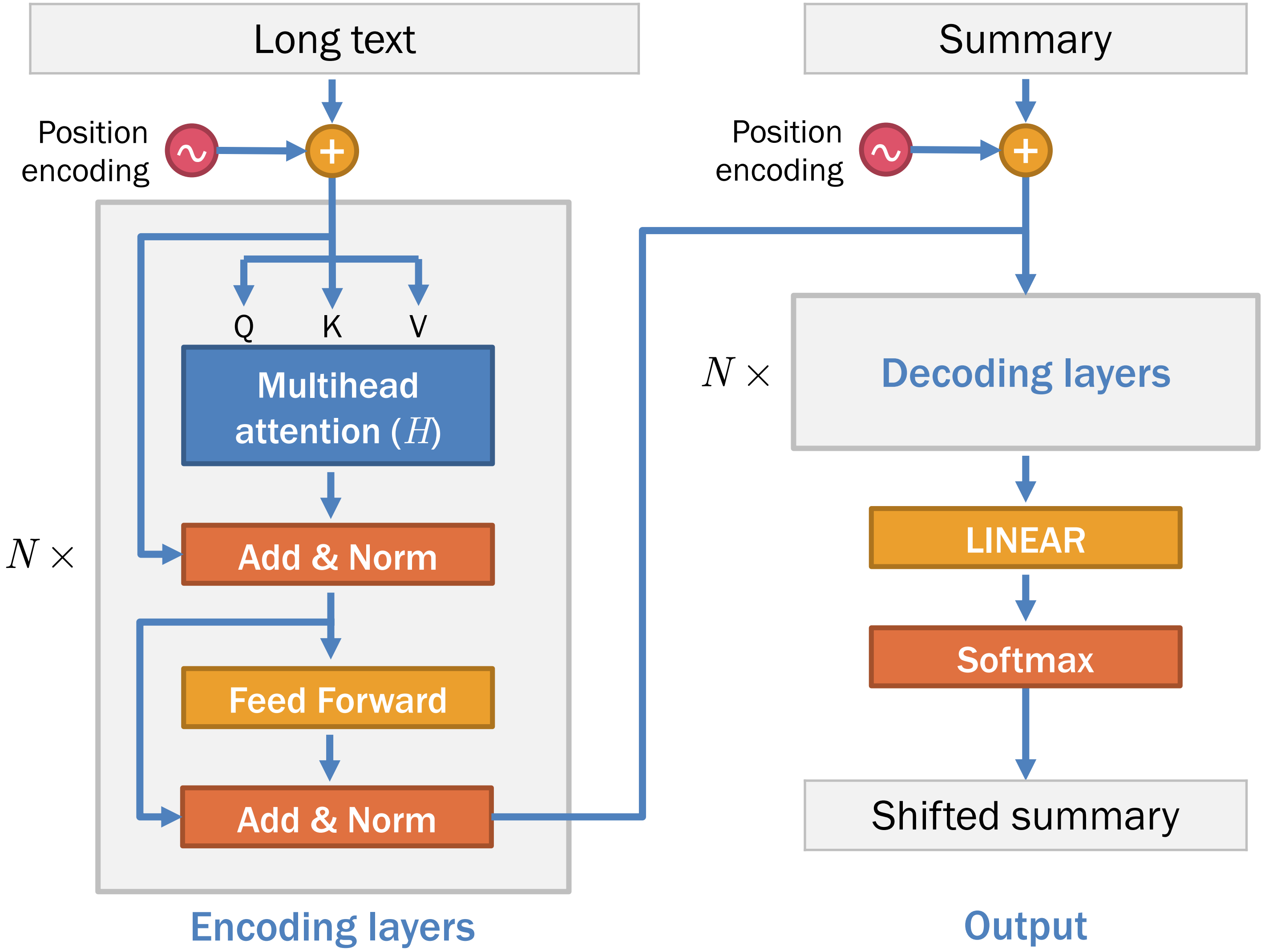
- BERT can be train via multiple downstream tasks
  - Machine translation
  - **Question answering (SQUAD)**
  - Inference in natural language (NLI in GLUE Dataset)
  - Abstractive summarization

Training BERT out of the Transformer



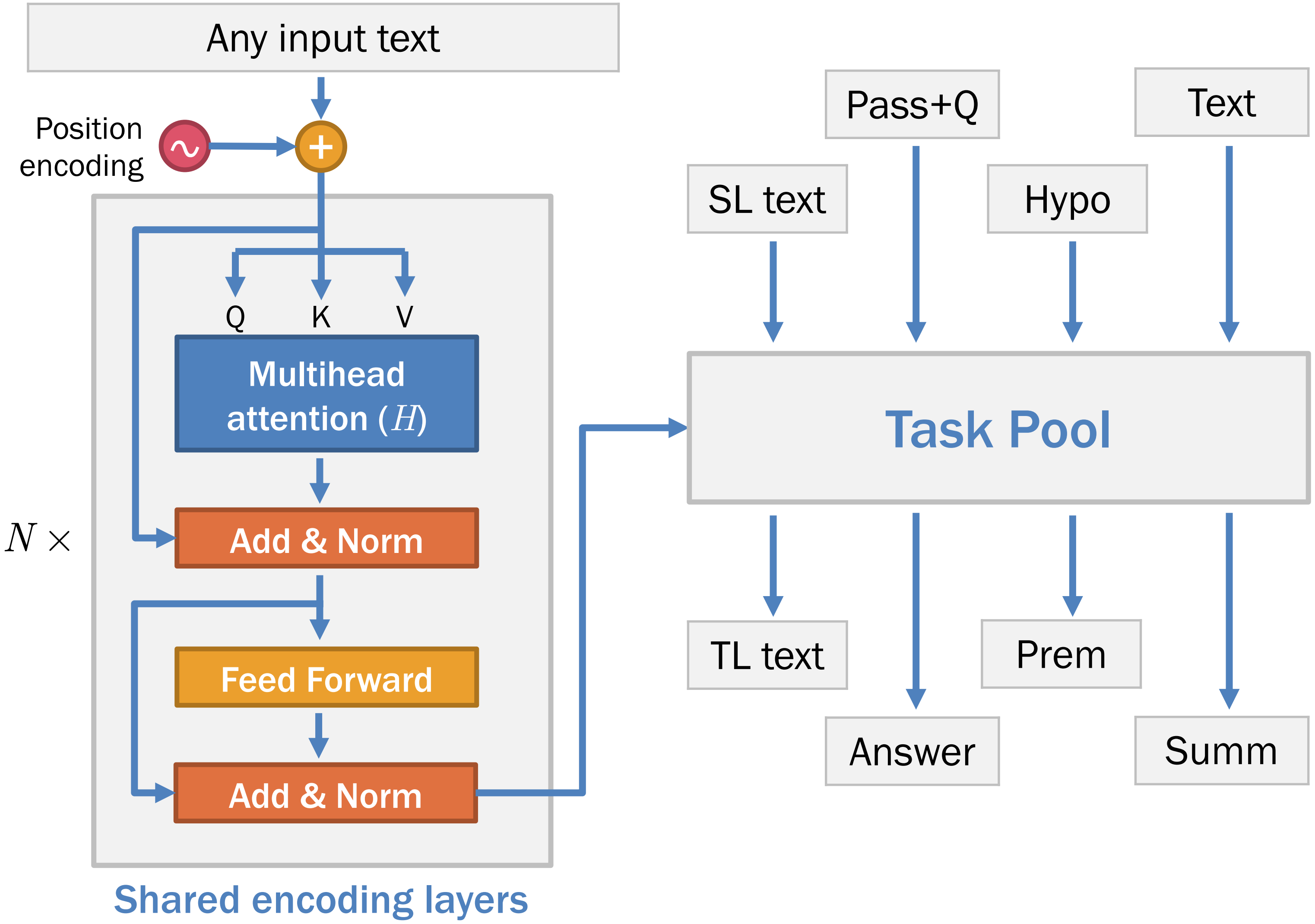
- BERT can be train via multiple downstream tasks
  - Machine translation
  - Question answering (SQUAD)
  - **Inference in natural language (NLI in GLUE Dataset)**
  - Abstractive summarization

Training BERT out of the Transformer



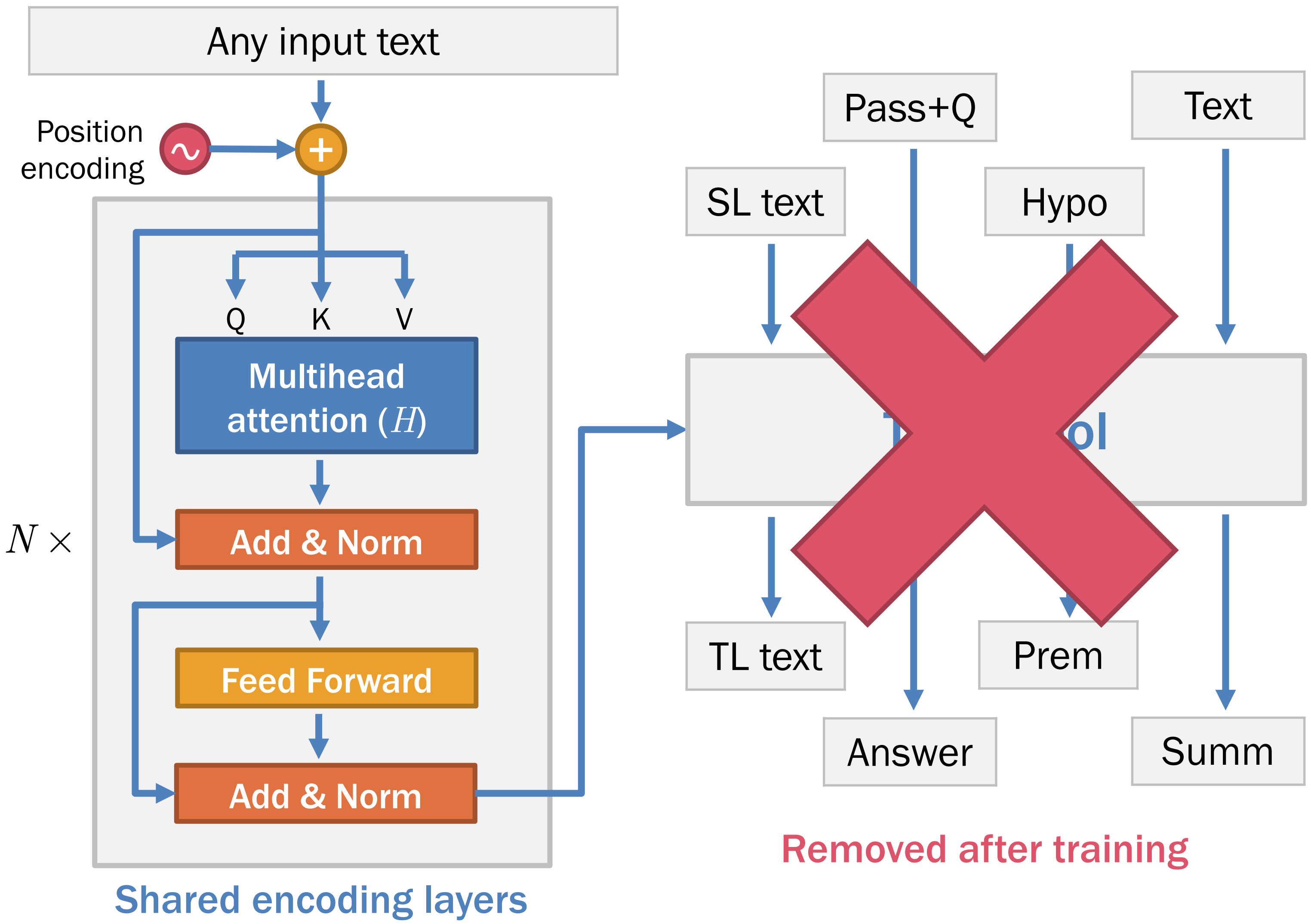
- BERT can be train via multiple downstream tasks
  - Machine translation
  - Question answering (SQUAD)
  - Inference in natural language (NLI in GLUE Dataset)
  - **Abstractive summarization**

Training BERT as Multitask Learning



- BERT can be train via multiple downstream tasks
  - Machine translation
  - Question answering (SQUAD)
  - Inference in natural language (NLI in GLUE Dataset)
  - Abstractive summarization

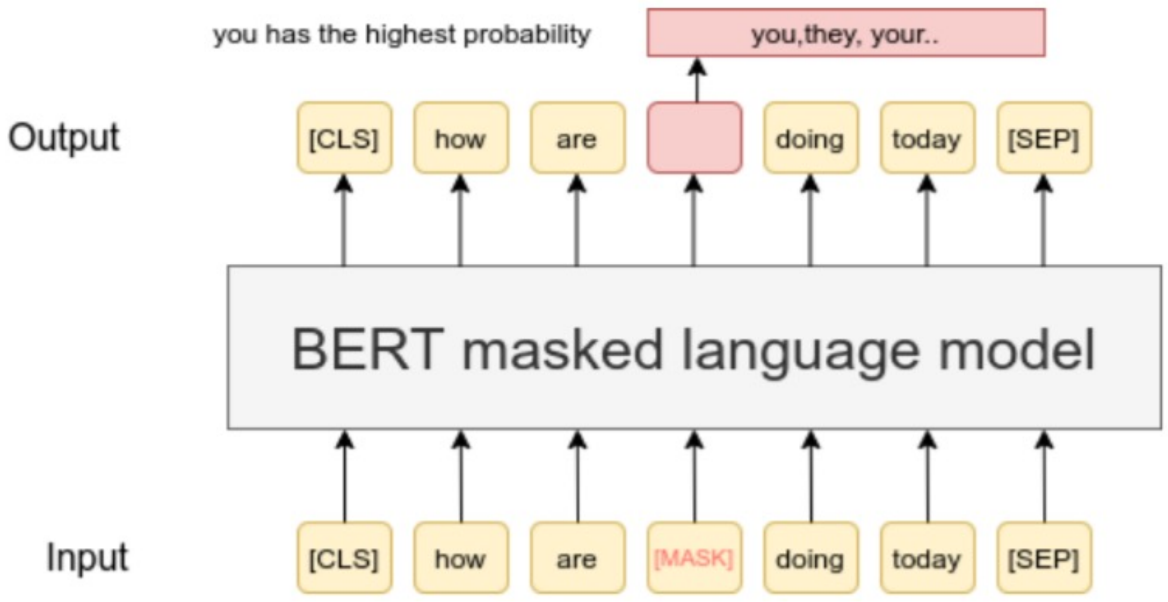
BERT = Shared Encoding Layers



- BERT can be train via multiple downstream tasks
  - Machine translation
  - Question answering (SQUAD)
  - Inference in natural language (NLI in GLUE Dataset)
  - Abstractive summarization

Masked Language Model (MLM)

- We replace some words in the input with blanks and compute the loss of word prediction on these blanks in the output

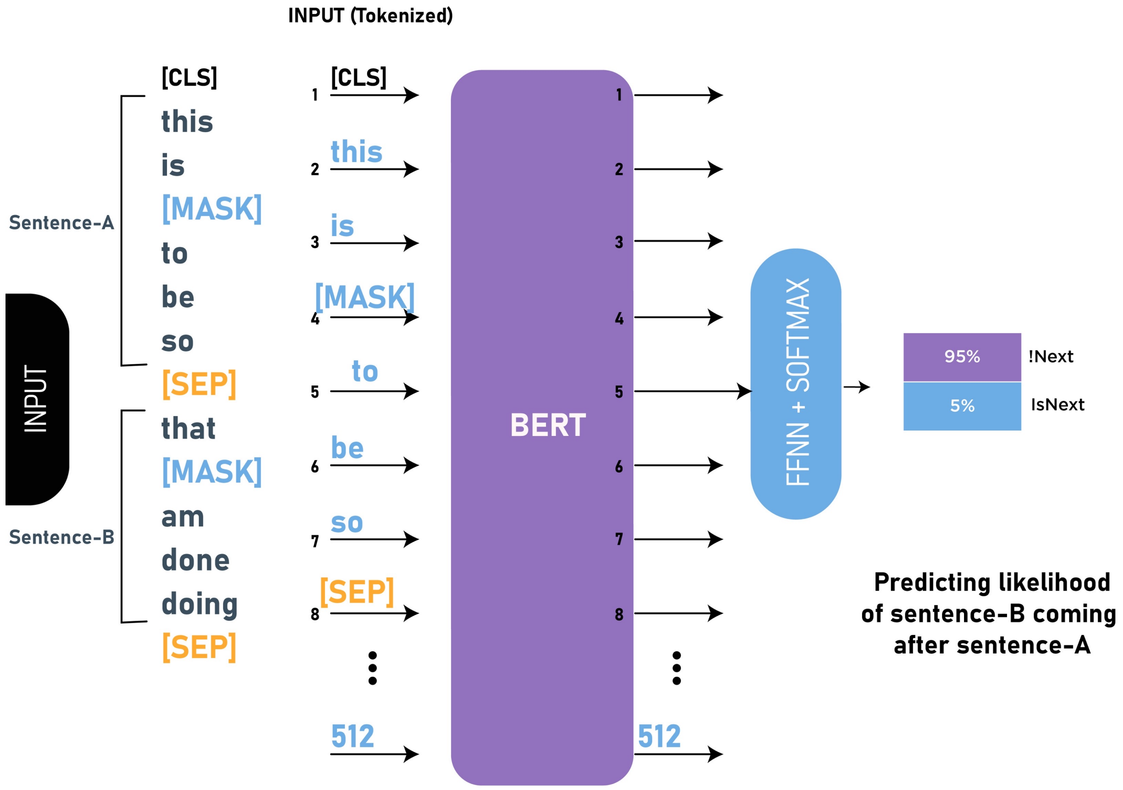


https://www.sbert.net/examples/unsupervised_learning/MLM/README.html

- Each input text is marked at some words by [MASK]
- Once marked, the masks will not be changed
- Special tokens
  - [CLS] = classifier token
  - [SEP] = separator token
  - [MASK] = mask

Next Sentence Prediction (NSP)

- We can concatenate two texts to let BERT learn their contextual information



https://www.geeksforgeeks.org/understanding-bert-nlp/

- In Natural Language Inference (NLI), each pair of sentences is classified as **entailment** or not (IsNext)
- With NSP training, semantic relatedness is imposed into word embedding

Cross-Lingual Language Model (XLM)

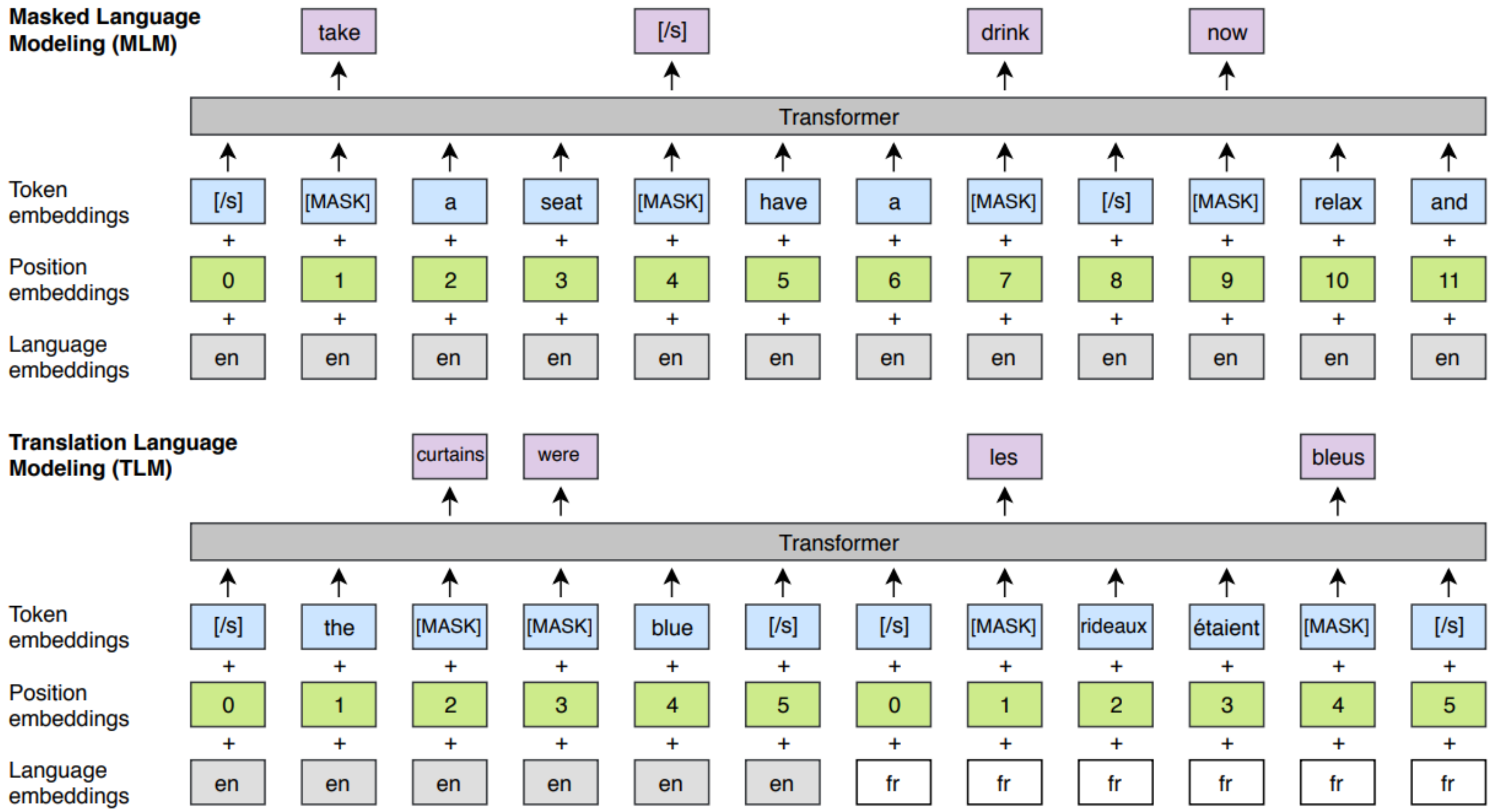


Figure 1: **Cross-lingual language model pretraining.** The MLM objective is similar to the one of Devlin et al. (2018), but with continuous streams of text as opposed to sentence pairs. The TLM objective extends MLM to pairs of parallel sentences. To predict a masked English word, the model can attend to both the English sentence and its French translation, and is encouraged to align English and French representations. Position embeddings of the target sentence are reset to facilitate the alignment.

https://bangliu.github.io/survey/2019/07/01/NLP-Pretraining/

- Translation pairs can also be used to train cross-lingual language model
- Some words are marked with [MASK] at random for masked language model
- Semantic relatedness can be learned from parallel corpora, especially from multitexts (multiple-language parallel texts)

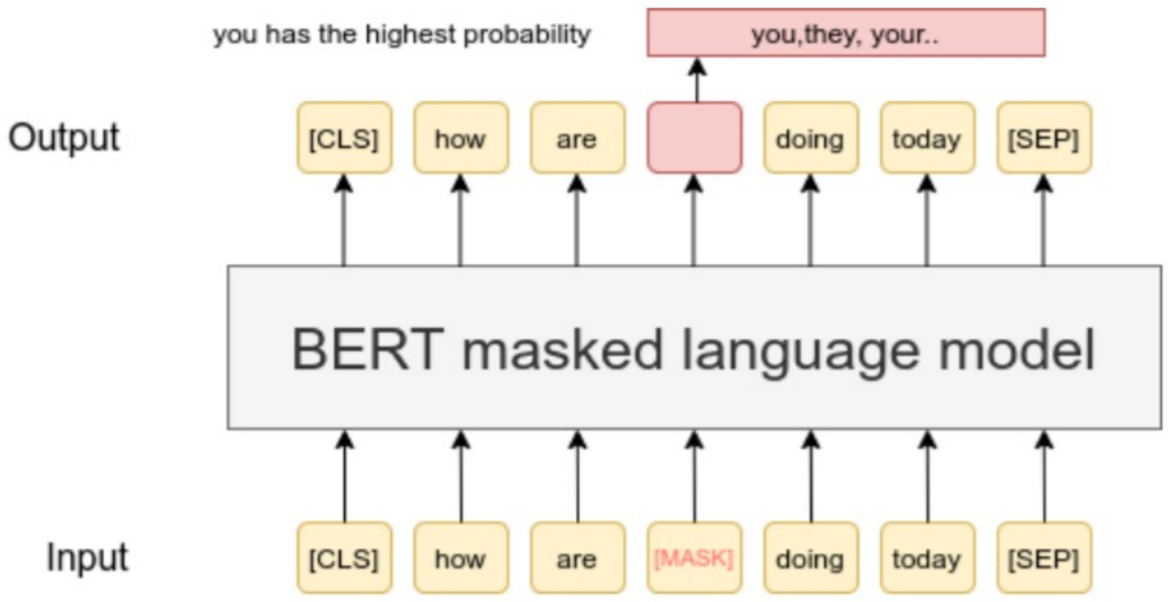
What Knowledge Does BERT Have?

- Syntactic knowledge
  - It encodes POS, idioms, and syntactic roles (Lin et al., 2019; Tenney et al., 2016; Liu et al., 2019)
  - It learns hierarchical idiomatic patterns; not syntax (Htut et al., 2019; Jawahar et al., 2019)
- Semantic knowledge
  - It encodes semantic roles (Ettinger, 2019) and entity types (Tenney et al., 2019)
  - It still struggles with representations of numbers (Wallace et al., 2019)
- World knowledge
  - It captures some commonsense knowledge (too many citations here)
  - It stuggles with pragmatic inference and role-based event knowledge (Ettinger, 2019)
  - It cannot still reason based on learned world knowledge (Forbes et al., 2019)

5. BERT Variants

RoBERTa (Liu et al., 2019)

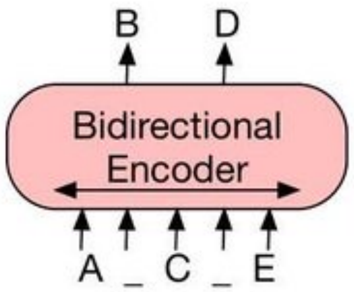
- Robustly Optimized BERT pretraining approach
  - An improved version of BERT



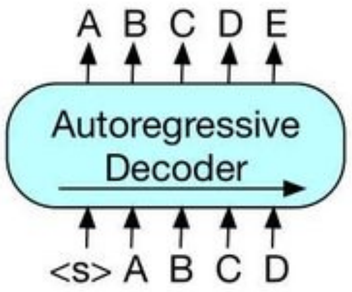
https://www.sbert.net/examples/unsupervised_learning/MLM/README.html

- Dynamic masking instead of static masking
- NSP task is eliminated without losing semantic relatedness
- Larger datasets are used in training than BERT (CC-News and Open WebText)

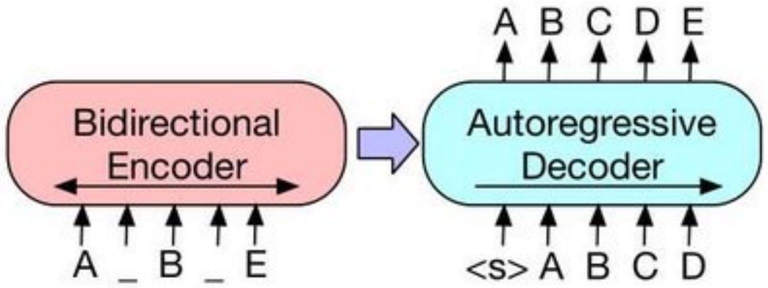
Differences of BERT, GPT, and BART (Lewis et al., 2019)



(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.



(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.

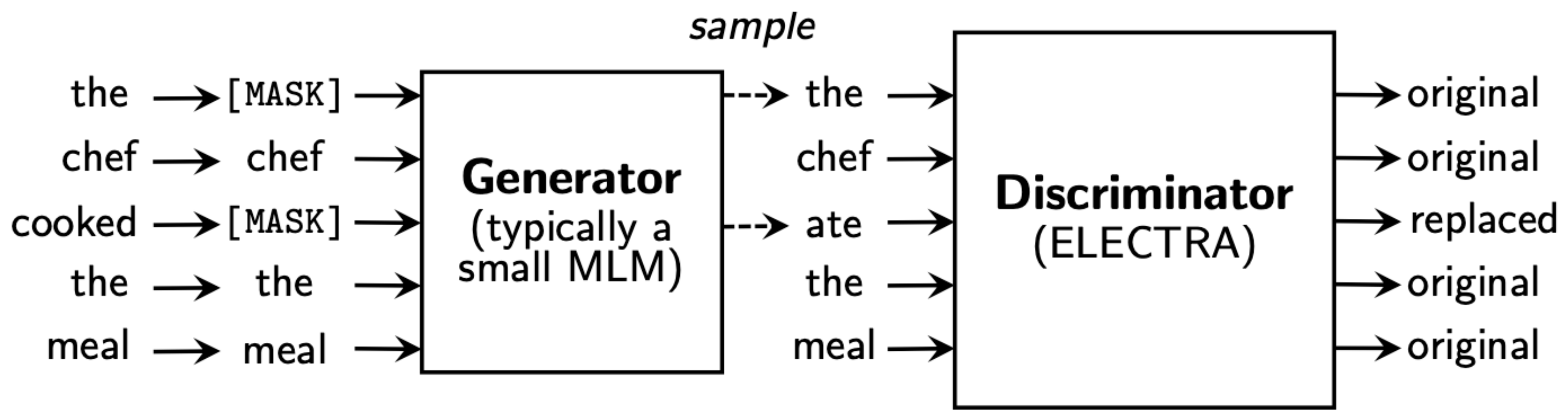


(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with a mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

- **BERT:** bidirectional encoder
- **GPT:** autoregressive (unidirectional) decoder
- **BART:** bidirectional encoder + autoregressive decoder

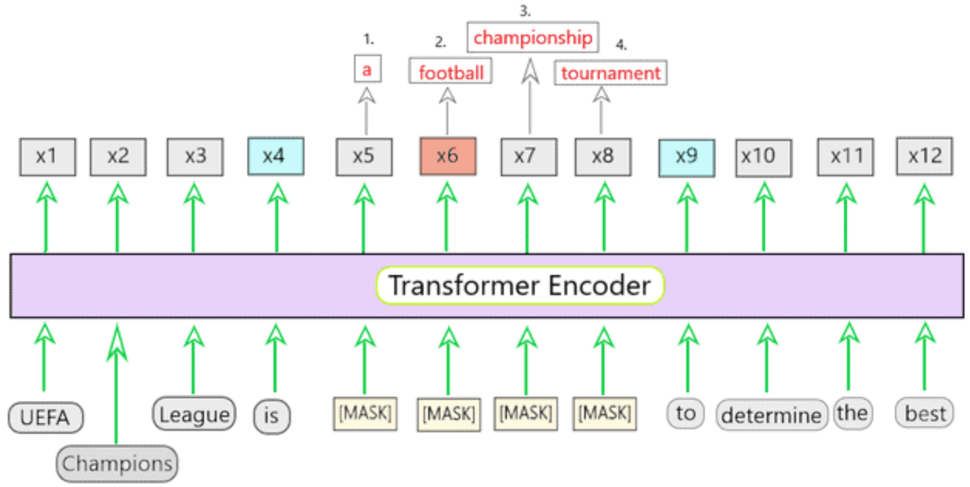
ELECTRA (Clark et al., 2020)

- Training by guessing the replaced tokens in the text
  - ELECTRA differs from BERT in that it is used as a discriminator
  - It is trained much faster and has much less parameters
  - It is frequently used in discriminative models

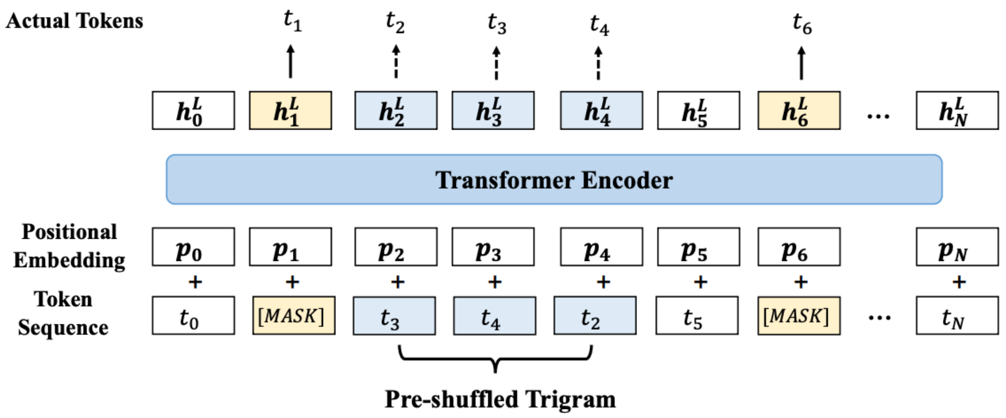


SpanBERT and StructBERT

- SpanBERT (Joshi et al., 2019): guess the missing chunk



- StructBERT (Wang et al., 2020): guess the right word order



(a) Word Structural Objective

T5 and mT5 (Raffel et al., 2020)

- T5 = Text-To-Text Transfer Transformer (five T's)

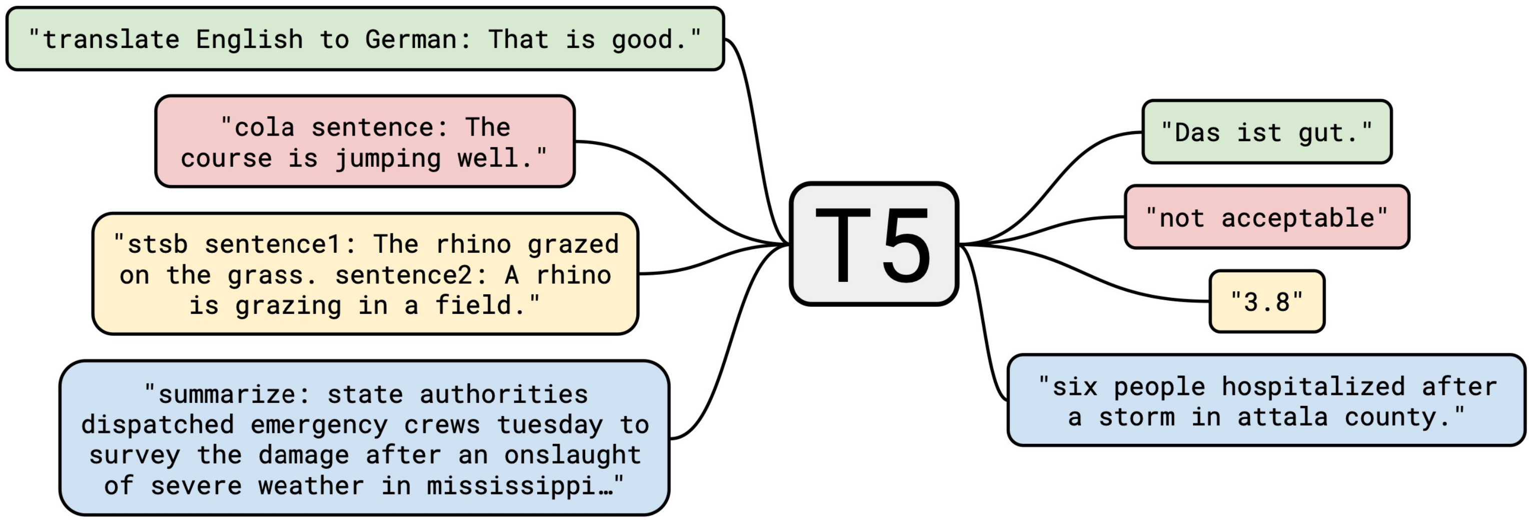
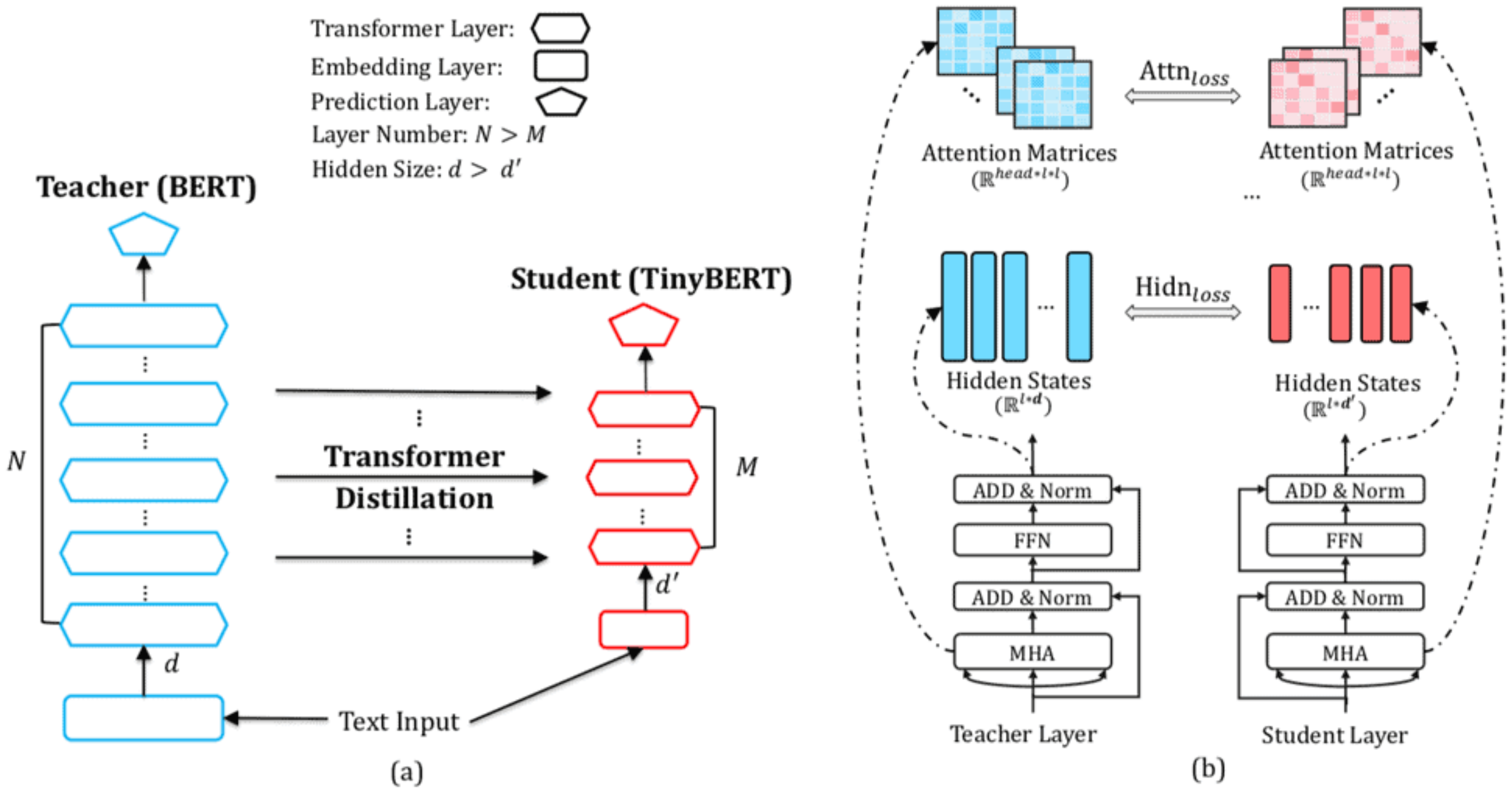


Figure 1: A diagram of our text-to-text framework. Every task we consider—including translation, question answering, and classification—is cast as feeding our model text as input and training it to generate some target text. This allows us to use the same model, loss function, hyperparameters, etc. across our diverse set of tasks. It also provides a standard testbed for the methods included in our empirical survey. "T5" refers to our model, which we dub the "**T**ext-**t**o-**T**ext **T**ransfer **T**ransformer".

- Every task is transformed into textual transfer
  - MT: "translate EN to DE"
  - Semantic similarity: "sim sent1: sent2:"
  - ATS: "summarize:"
- mT5 is a multilingual version of T5 model

DistilBERT (Sanh et al., 2020)

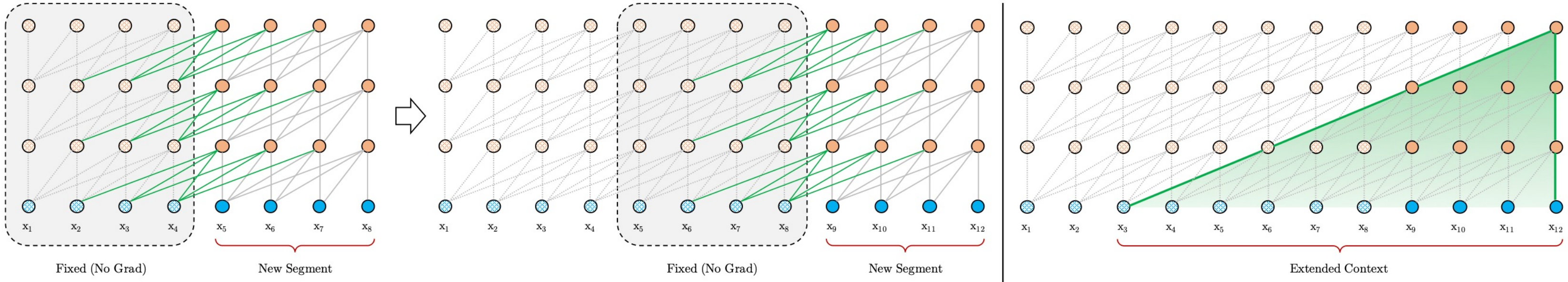
- Knowledge distillation from a very large model to a comparable, small model



- Imitating how the large model works by enforcing the losses of hidden states and attention matrices

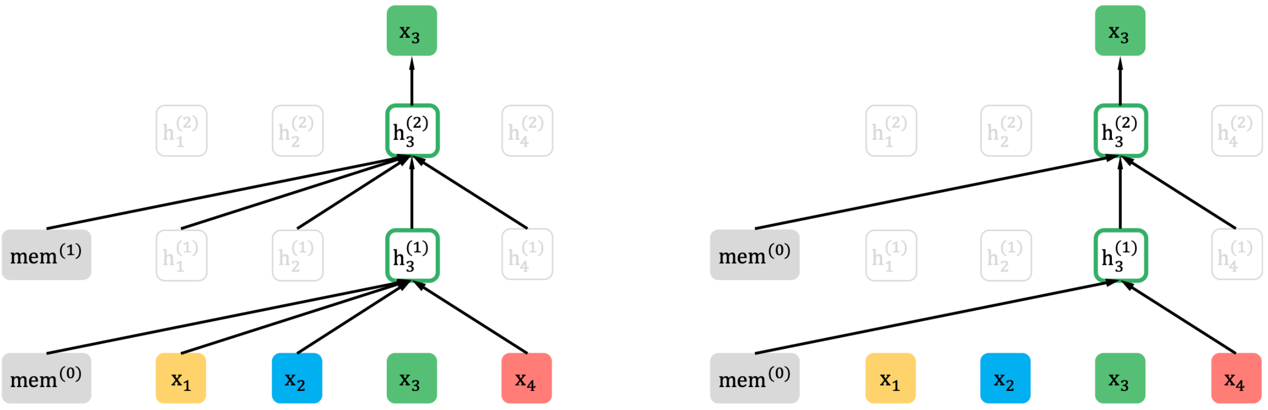
XLNet (Yang et al., 2019)

- Cope with large input sequences with Transformer-XL



(a) Training phase. (b) Evaluation phase.

- Permutation language model
  - Flexible bidirectional context
  - Ex. $1 \rightarrow 4 \rightarrow 2 \rightarrow 3$ means $P(x_1) + P(x_4 \mid x_1) + P(x_2 \mid x_1, x_4) + P(x_3 \mid x_1, x_4, x_2)$



Factorization order: 1 → 4 → 2 → 3   Factorization order: 4 → 3 → 1 → 2

6. Conclusion and Discussion Time

Conclusion

- The Transformer model is a sequence-to-sequence model
  - It learns to encode phrase structures in the source sequence in the self-attention
  - It learns to align phrase structures in the source to the target sequence using the alignment attention
  - It learns to produce a target sequence using next-word prediction from the encoded phrase structures
- Upper bounds
  - **Encoder:** H^N non-recursive phrases
  - **Decoder:** $H^{N_\mathrm{E}+N_\mathrm{D}}$ non-recursive translation pairs

Neural Machine Translation

- How many phrases and translation pairs can we extract from the dataset?
- Can we list up all translation pairs learned by the Transformer model?
- What is the longest phrase that can be reordered correctly?
- How can we circumvent the issues of over-generation and under-generation?

Abstractive Summarization

- How many proper names and multiword expressions are there in the dataset?
- How can the topic sentence be detected?
- Can we list up all conceptualization rules learned by the Transformer model?
- What is the longest phrase that is conceptualized into 1-5 words?
- How can we circumvent the issues of under-generation (more frequent) and over-generation (less frequent)?

Thank You