Transformers are Graph Neural Networks

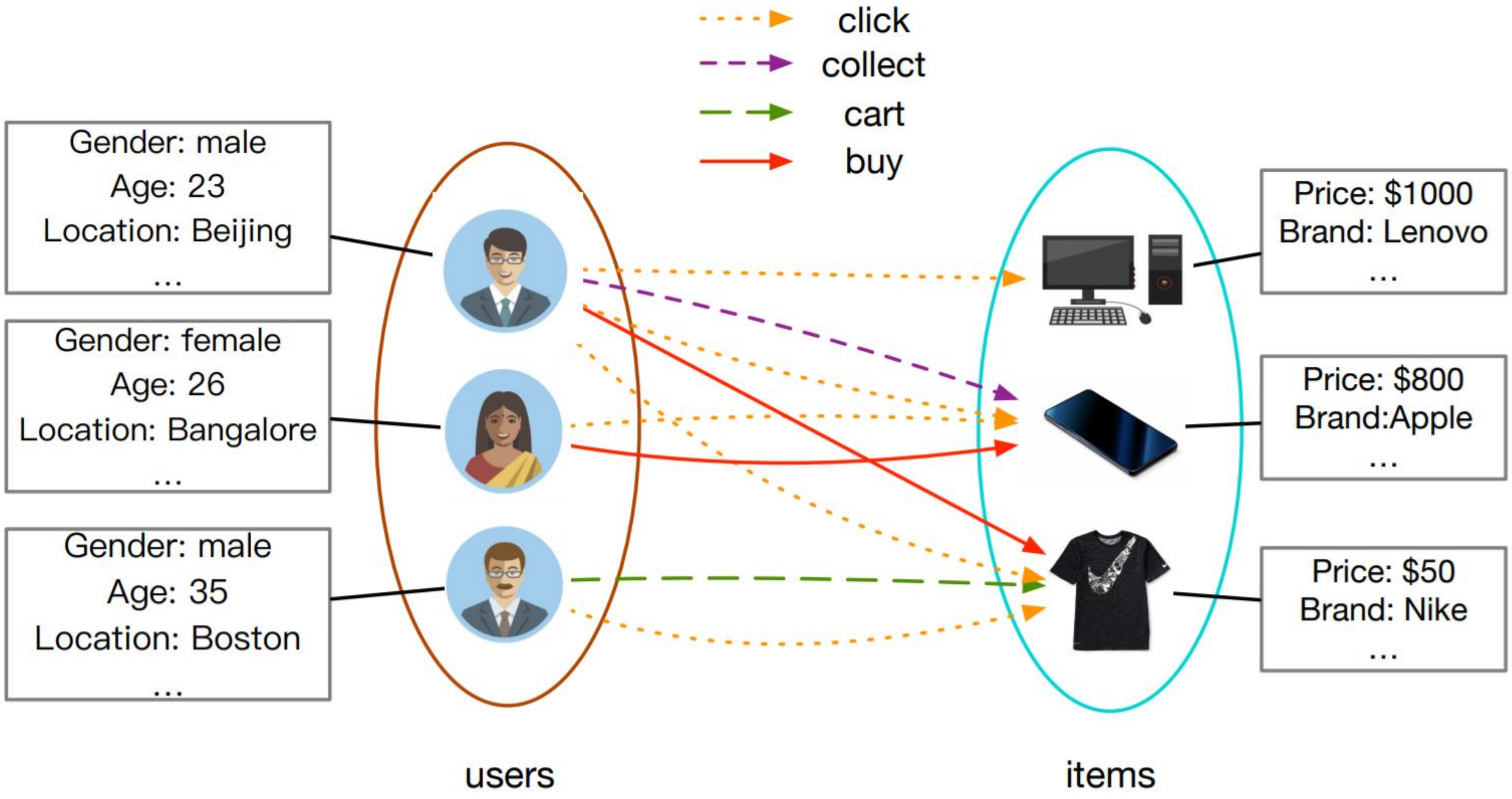
Chaitanya K. Joshi

Graph Deep Learning Reading Group

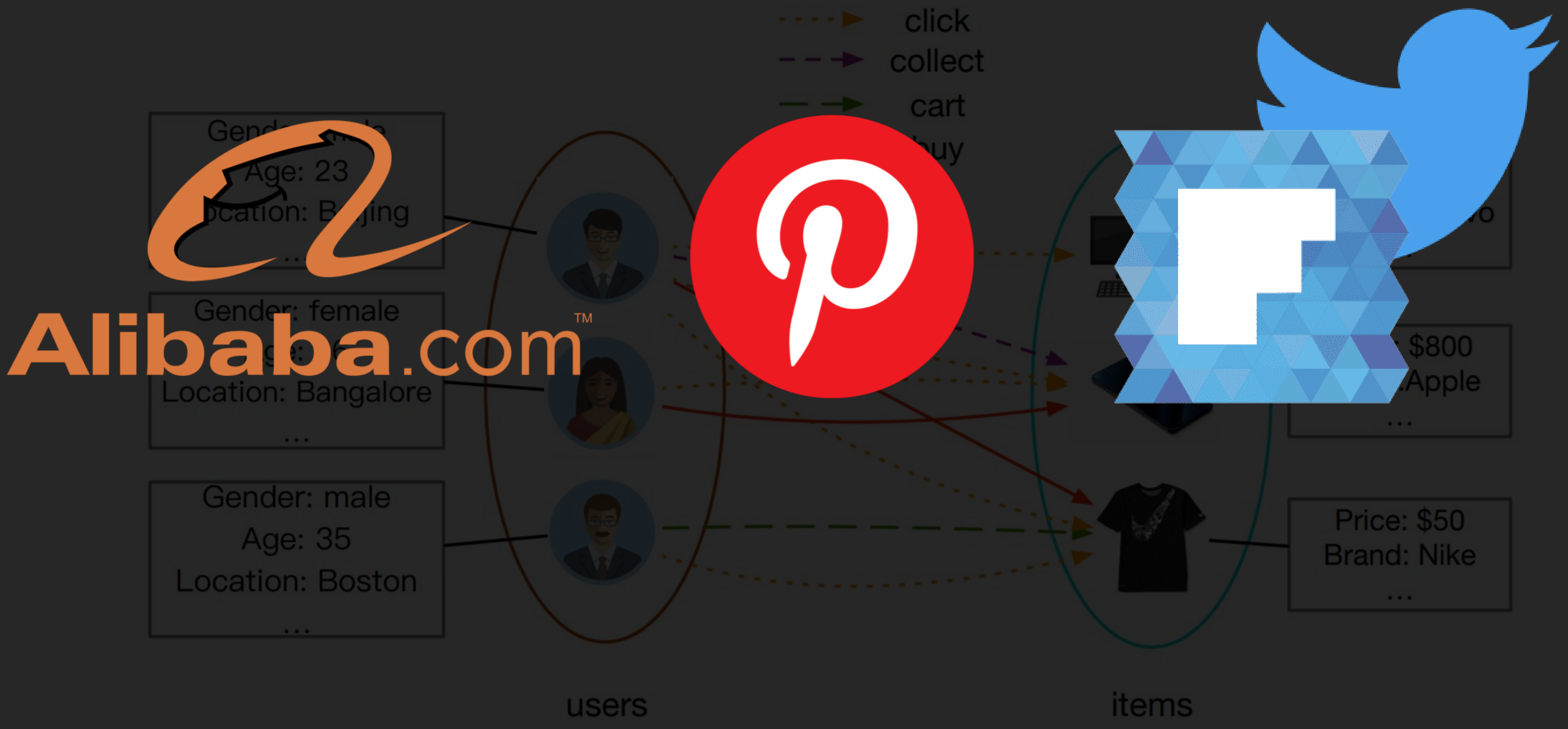
Full Blogpost:

https://graphdeeplearning.github.io/post/transformers-are-gnns/

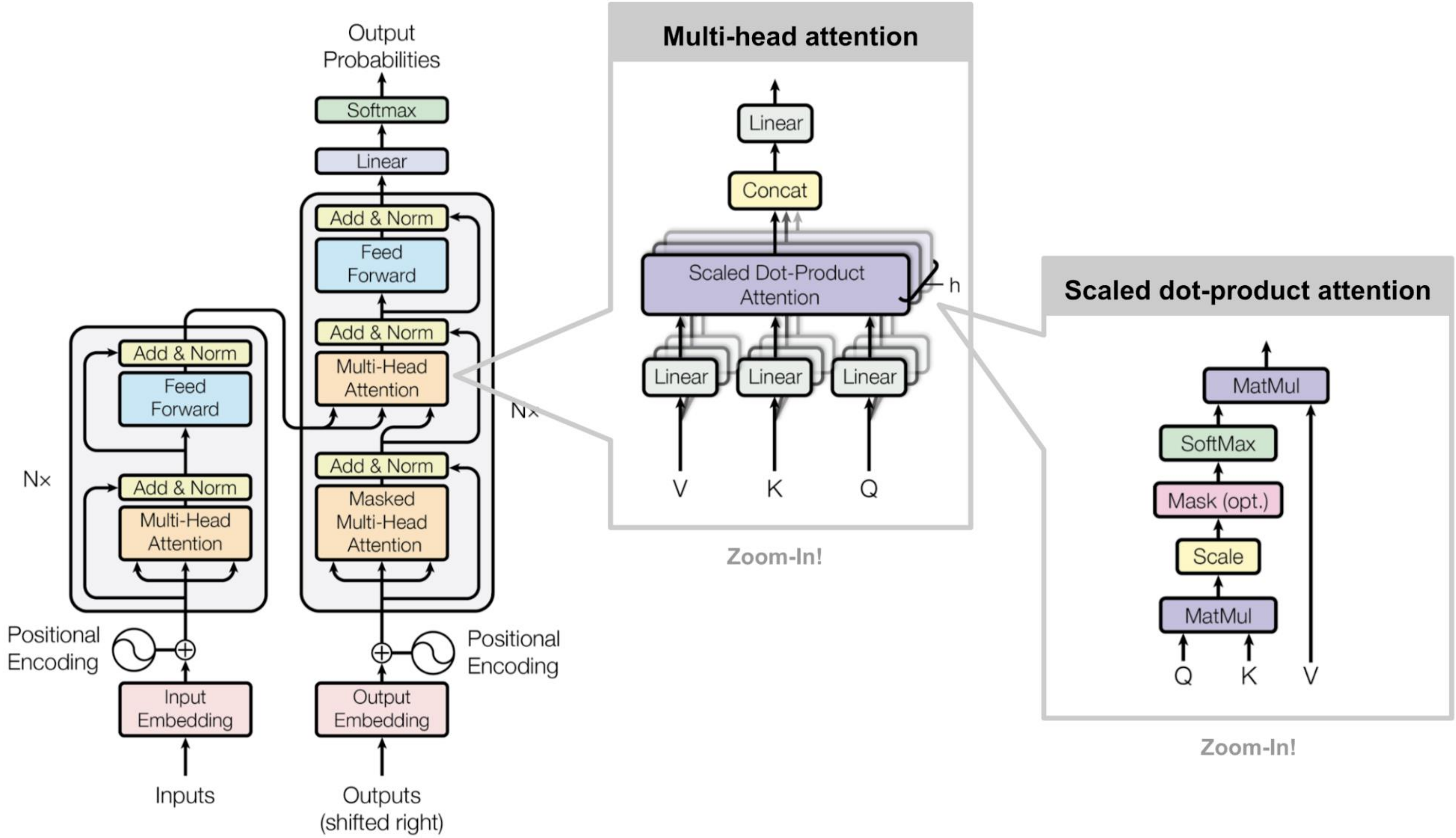
Some success stories: GNNs for RecSys

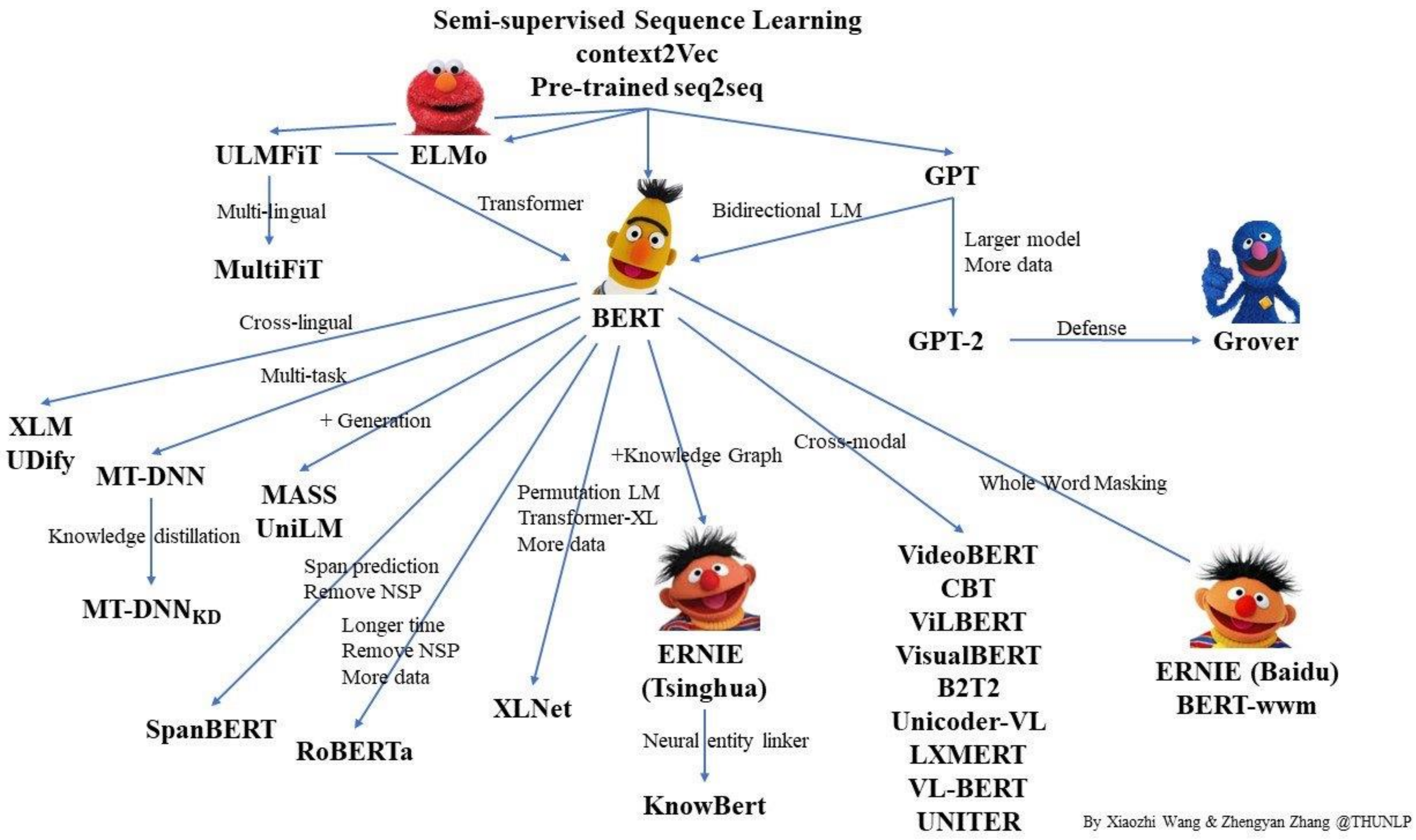


Some success stories: GNNs for RecSys



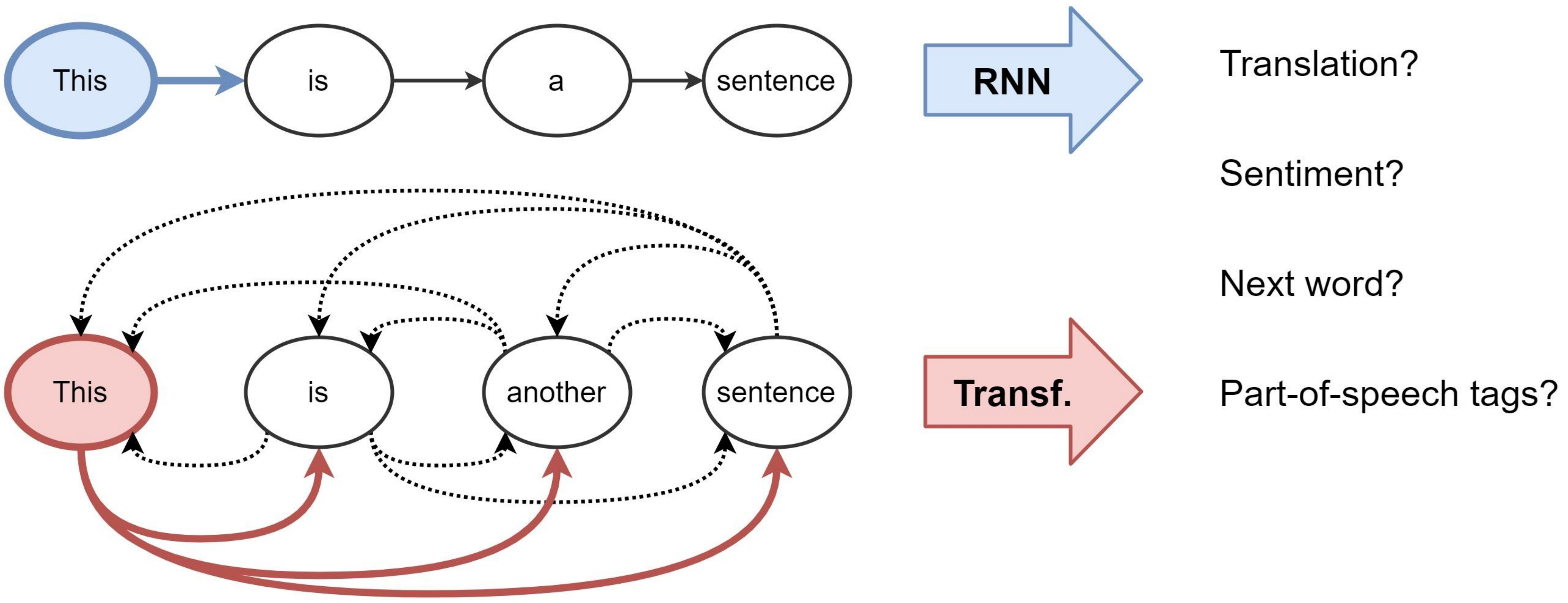
Another success story?
The Transformer architecture for NLP





Semi-supervised Sequence Learning
context2Vec
Pre-trained seq2seq
ULMFiT
ELMo
GPT
Transformer
Bidirectional LM
Multi-lingual
MultiFiT
Larger model
More data
BERT
Cross-lingual
GPT-2
Defense
Grover
Multi-task
XLM
UDify
+ Generation
MT-DNN
MASS
UniLM
+Knowledge Graph
Cross-modal
Whole Word Masking
Permutation LM
Transformer-XL
More data
Knowledge distillation
Span prediction
Remove NSP
MT-DNN$_{KD}$
Longer time
Remove NSP
More data
VideoBERT
CBT
ViLBERT
VisualBERT
B2T2
Unicoder-VL
LXMERT
VL-BERT
UNITER
ERNIE (Tsinghua)
ERNIE (Baidu)
BERT-wwm
SpanBERT
RoBERTa
XLNet
Neural entity linker
KnowBert
By Xiaozhi Wang & Zhengyan Zhang @THUNLP

Representation Learning for NLP



Breaking down the Transformer

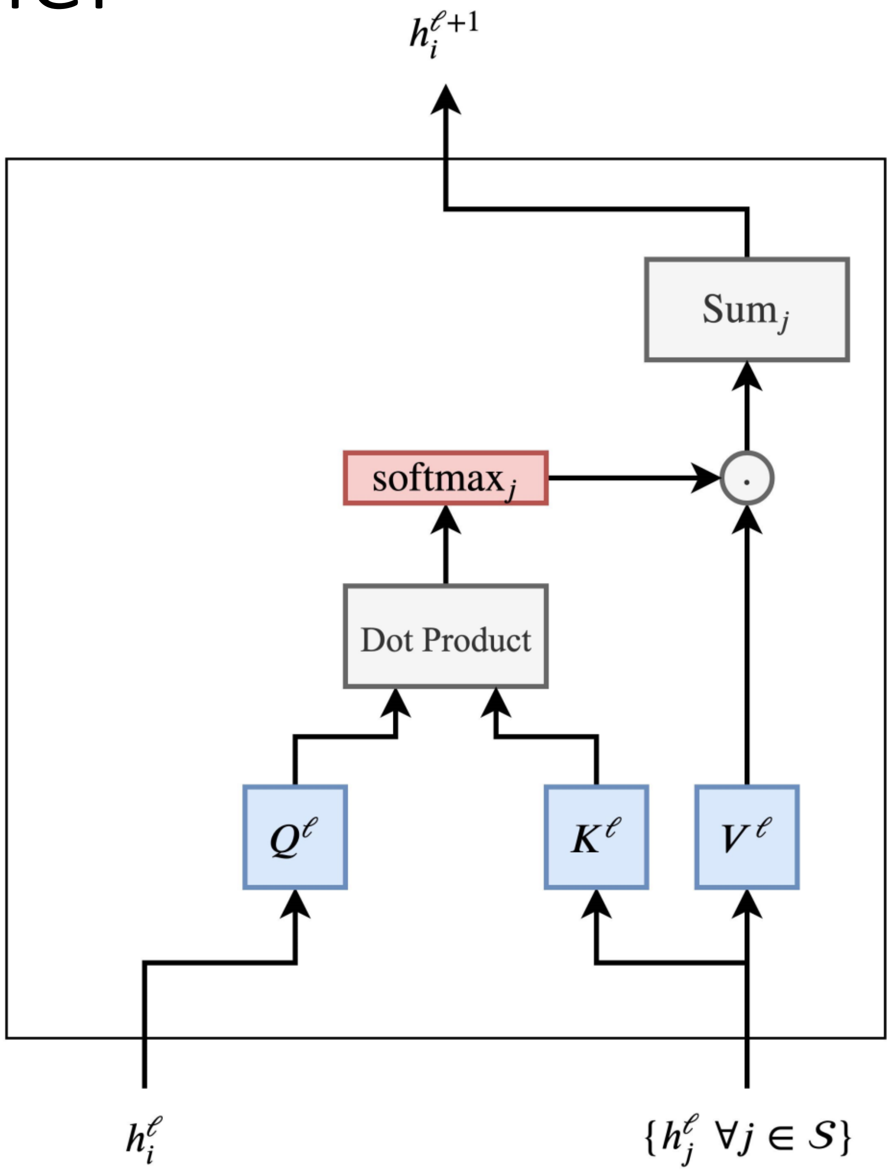
We update the hidden feature h of the i'th word in a sentence S from layer ℓ to layer ℓ+1 as follows:

$$h_i^{\ell+1} = \text{Attention}\big(Q^\ell h_i^\ell\ , K^\ell h_j^\ell\ , V^\ell h_j^\ell\big),$$

$$i.\ e.\ ,\ h_i^{\ell+1} = \sum_{j \in \mathcal{S}} w_{ij}\big(V^\ell h_j^\ell\big),$$

$$\text{where } w_{ij} = \text{softmax}_j\big(Q^\ell h_i^\ell \cdot K^\ell h_j^\ell\big),$$

where $j \in S$ denotes the set of words in the sentence and Q, K, V are learnable linear weights.



Breaking down the Transformer

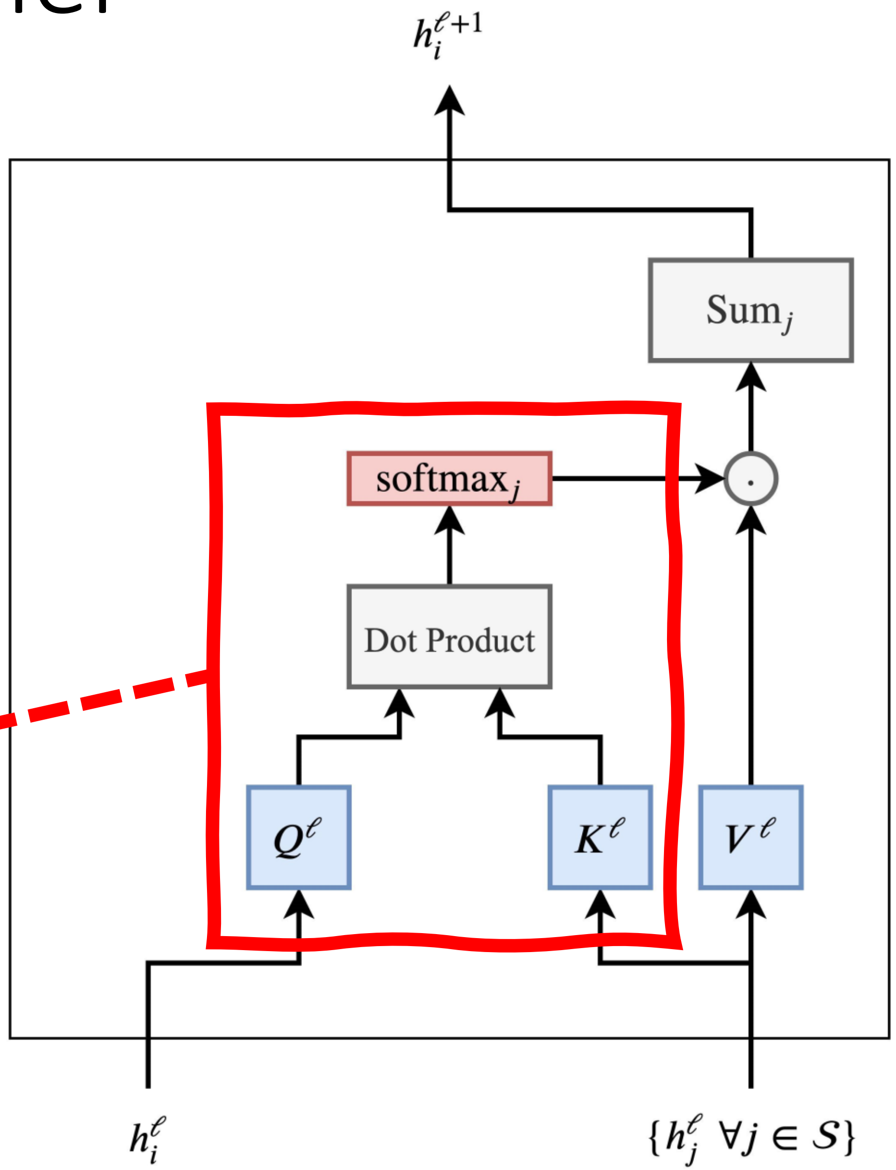
We update the hidden feature h of the i'th word in a sentence S from layer ℓ to layer $\ell+1$ as follows:

$$h_i^{\ell+1} = \text{Attention}\left(Q^\ell h_i^\ell \text{ , } K^\ell h_j^\ell \text{ , } V^\ell h_j^\ell\right),$$

$$i.\ e.\ ,\ h_i^{\ell+1} = \sum_{j \in \mathcal{S}} w_{ij} \left(V^\ell h_j^\ell\right),$$

$$\text{where } \boxed{w_{ij} = \text{softmax}_j \left(Q^\ell h_i^\ell \cdot K^\ell h_j^\ell\right),}$$

where $j \in S$ denotes the set of words in the sentence and Q, K, V are learnable linear weights.



Breaking down the Transformer

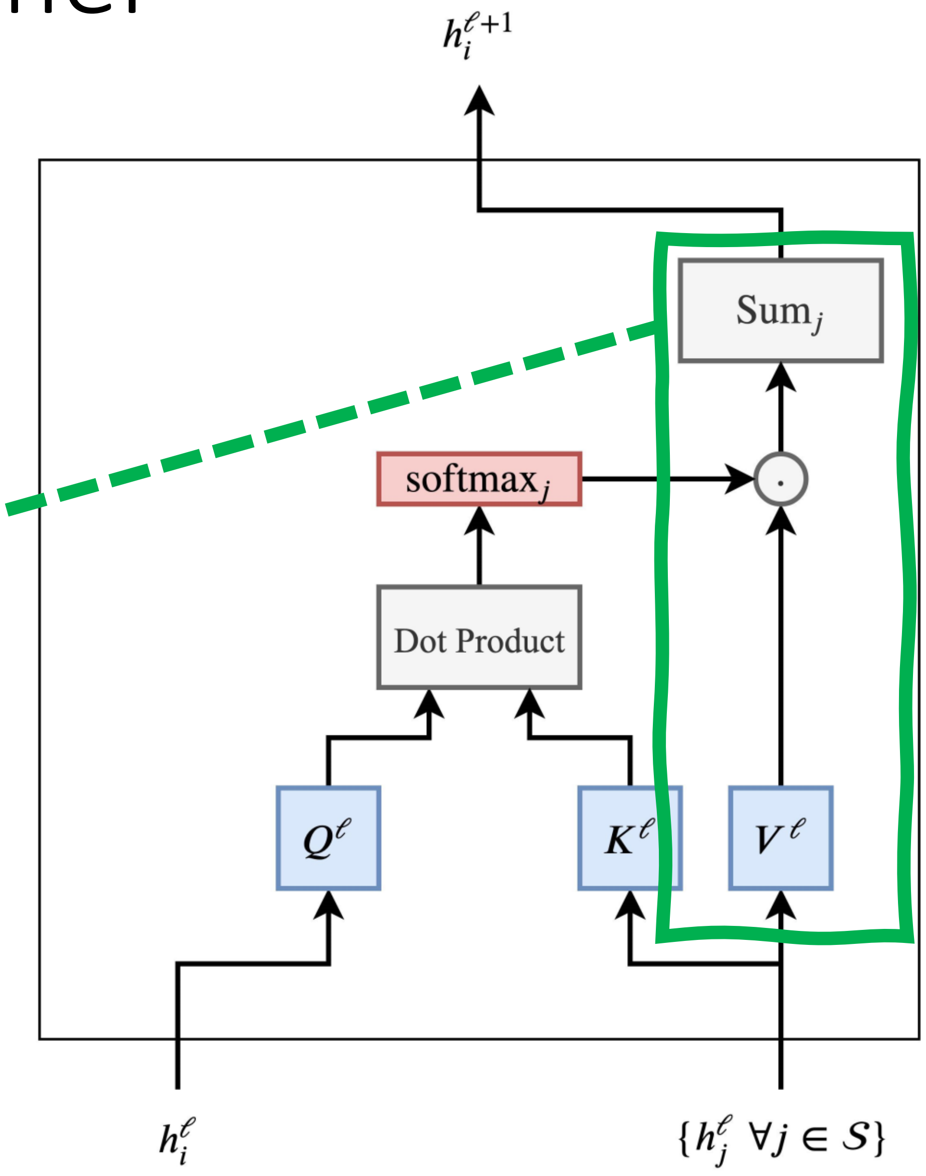
We update the hidden feature h of the i'th word in a sentence S from layer ℓ to layer $\ell+1$ as follows:

$$h_i^{\ell+1} = \text{Attention}\left(Q^\ell h_i^\ell \ , K^\ell h_j^\ell \ , V^\ell h_j^\ell\right),$$

$$i.\ e.\ \boxed{h_i^{\ell+1} = \sum_{j \in \mathcal{S}} w_{ij} \left(V^\ell h_j^\ell\right),}$$

$$\text{where } w_{ij} = \text{softmax}_j\left(Q^\ell h_i^\ell \cdot K^\ell h_j^\ell\right),$$

where $j \in S$ denotes the set of words in the sentence and Q, K, V are learnable linear weights.



Multi-head Attention

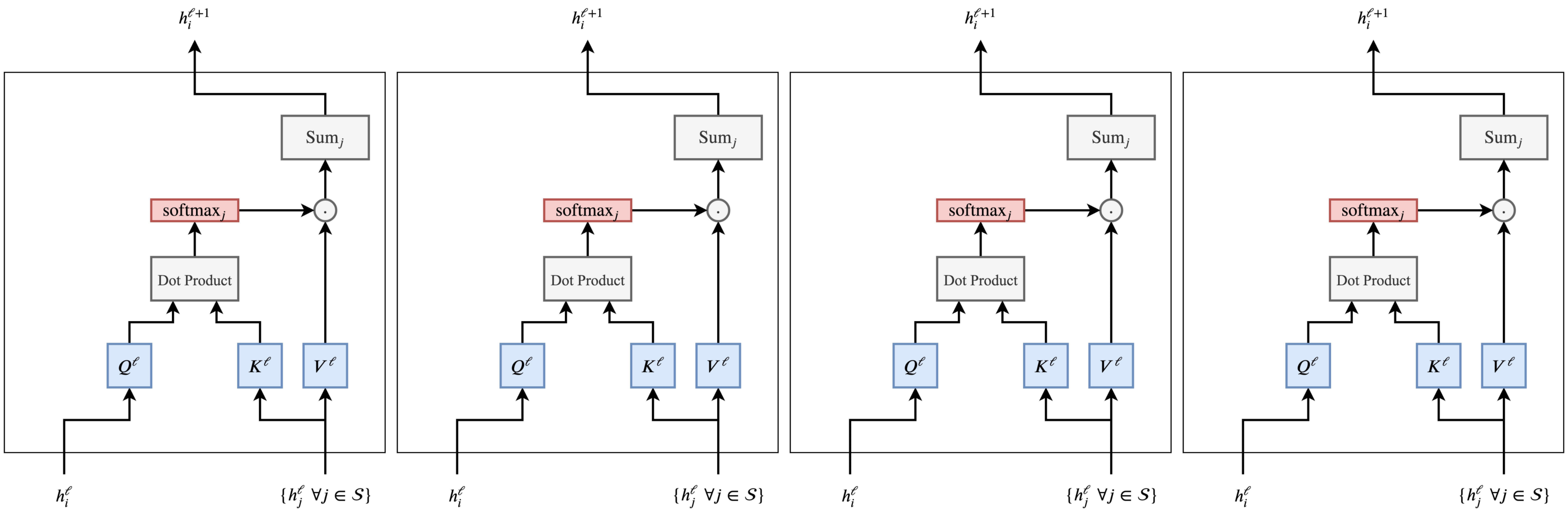
Bad random initializations can de-stabilize the learning process of this dot-product attention mechanism. We can 'hedge our bets' through concatenating multiple attention 'heads':

$$h_i^{\ell+1} = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_K)\, O^{\ell},$$

$$\mathrm{head}_k = \mathrm{Attention}\left(Q^{k,\ell} h_i^{\ell}\,,\, K^{k,\ell} h_j^{\ell}\,,\, V^{k,\ell} h_j^{\ell}\right),$$

Multi-head Attention

$$h_i^{\ell+1} = \mathrm{Concat}(\overbrace{\mathrm{head}_1, \ldots, \mathrm{head}_K}) O^{\ell},$$



The Final Picture

Update each word's features through

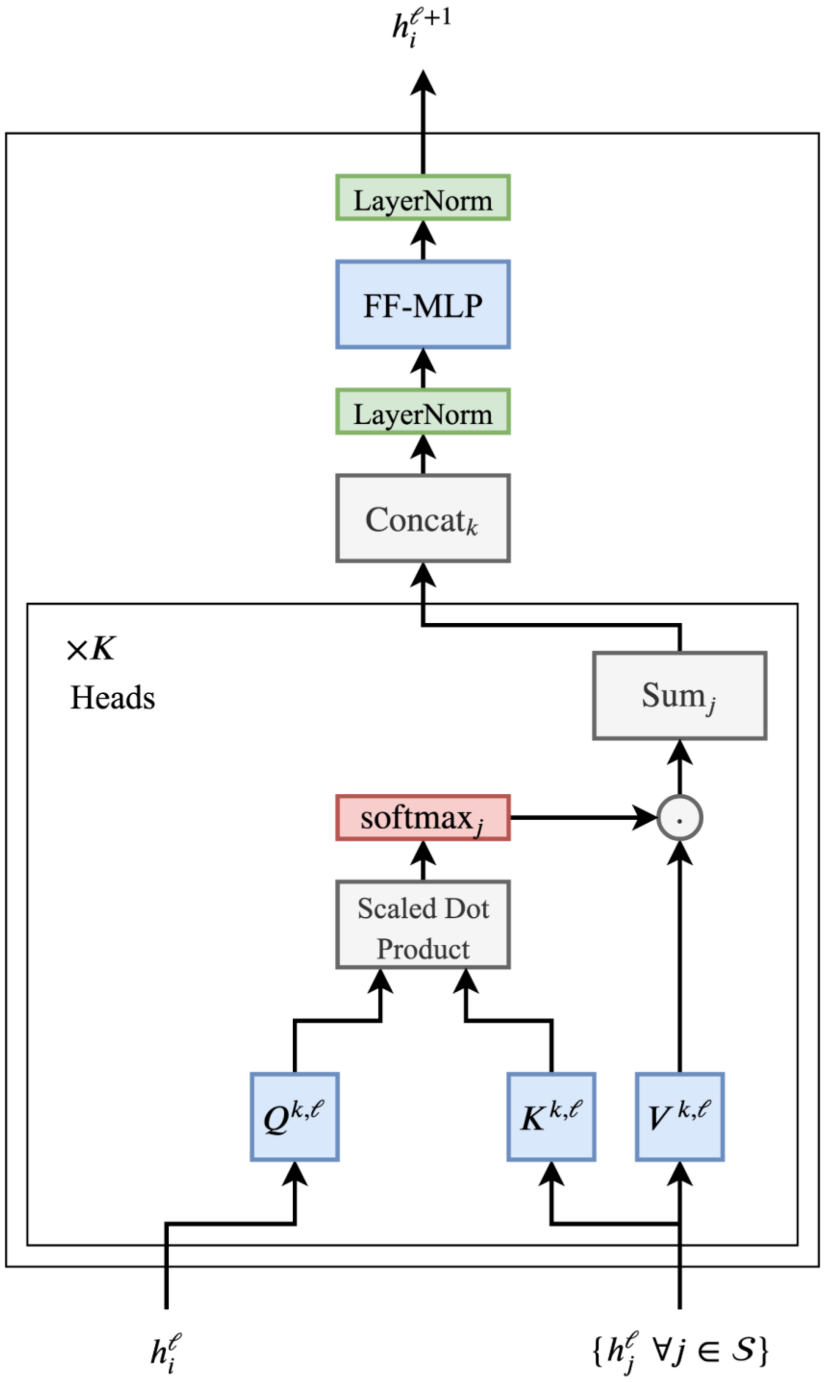
Multi-head Attention mechanism

as a weighted sum of features of other words in the sentence.

+ Scaling dot product attention

+ Normalization layers

+ Residual links



The Final Picture

Update each word's features through

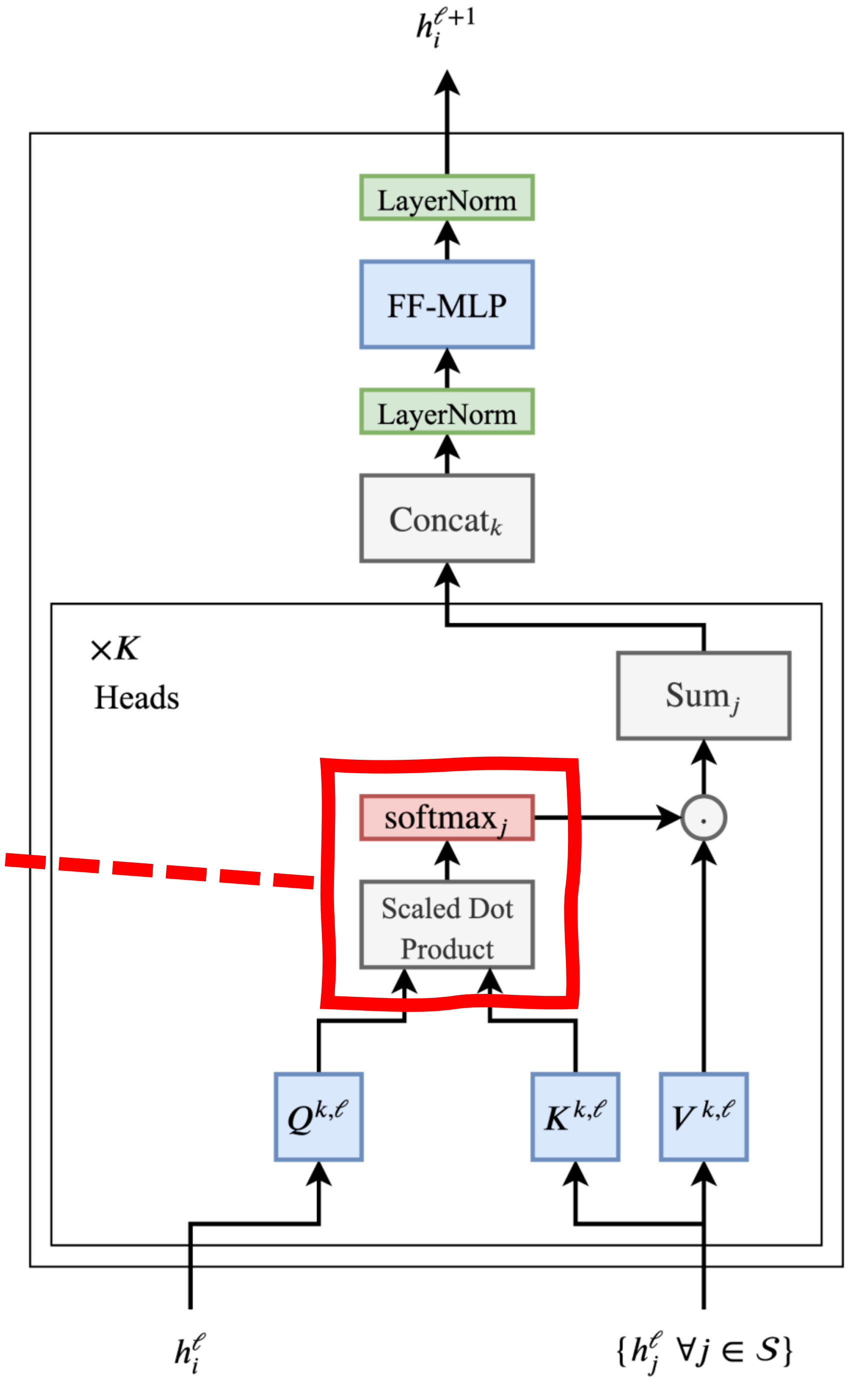
Multi-head Attention mechanism

as a weighted sum of features of other words in the sentence.

+ Scaling dot product attention

+ Normalization layers

+ Residual links



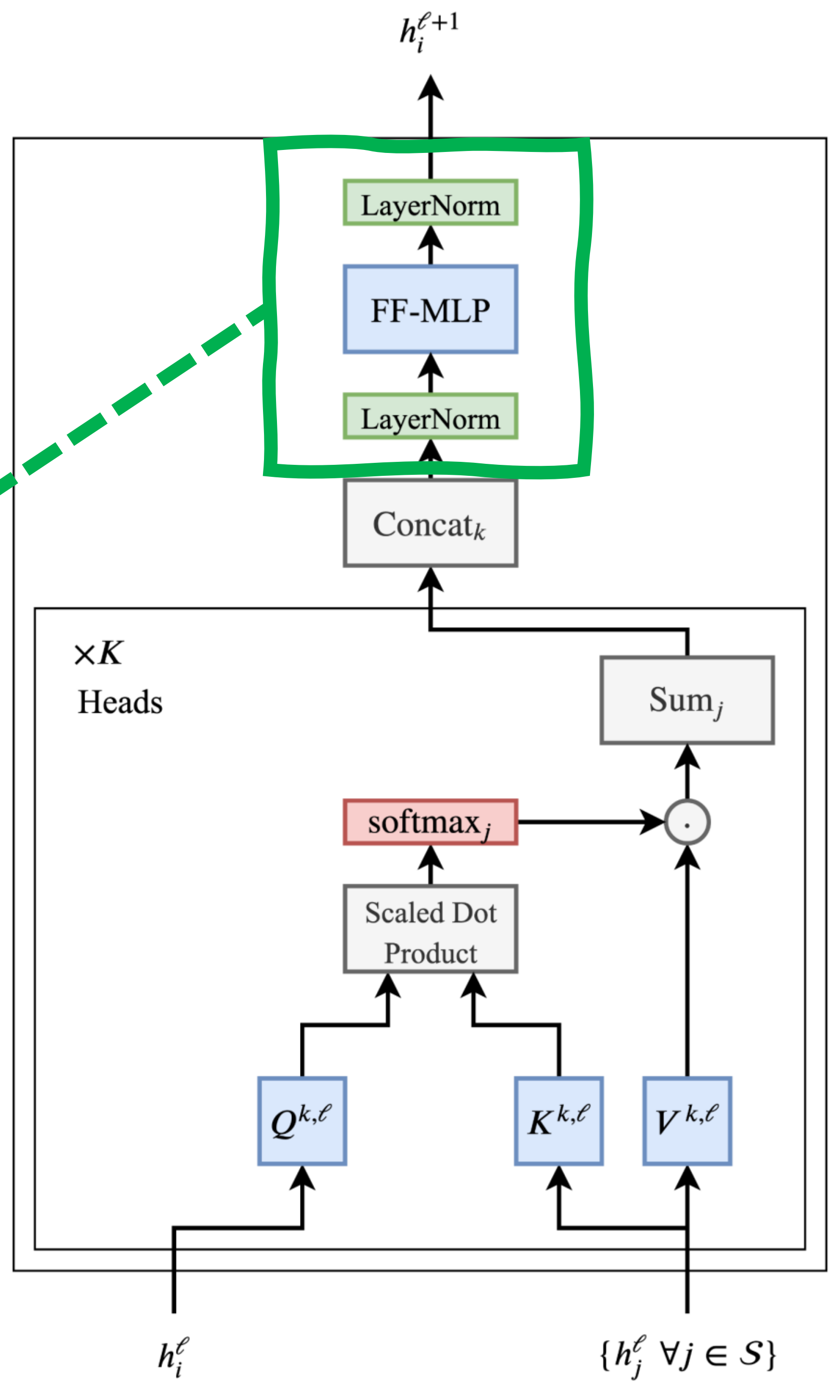
The Final Picture

Update each word's features through
Multi-head Attention mechanism
as a weighted sum of features of other words in the sentence.

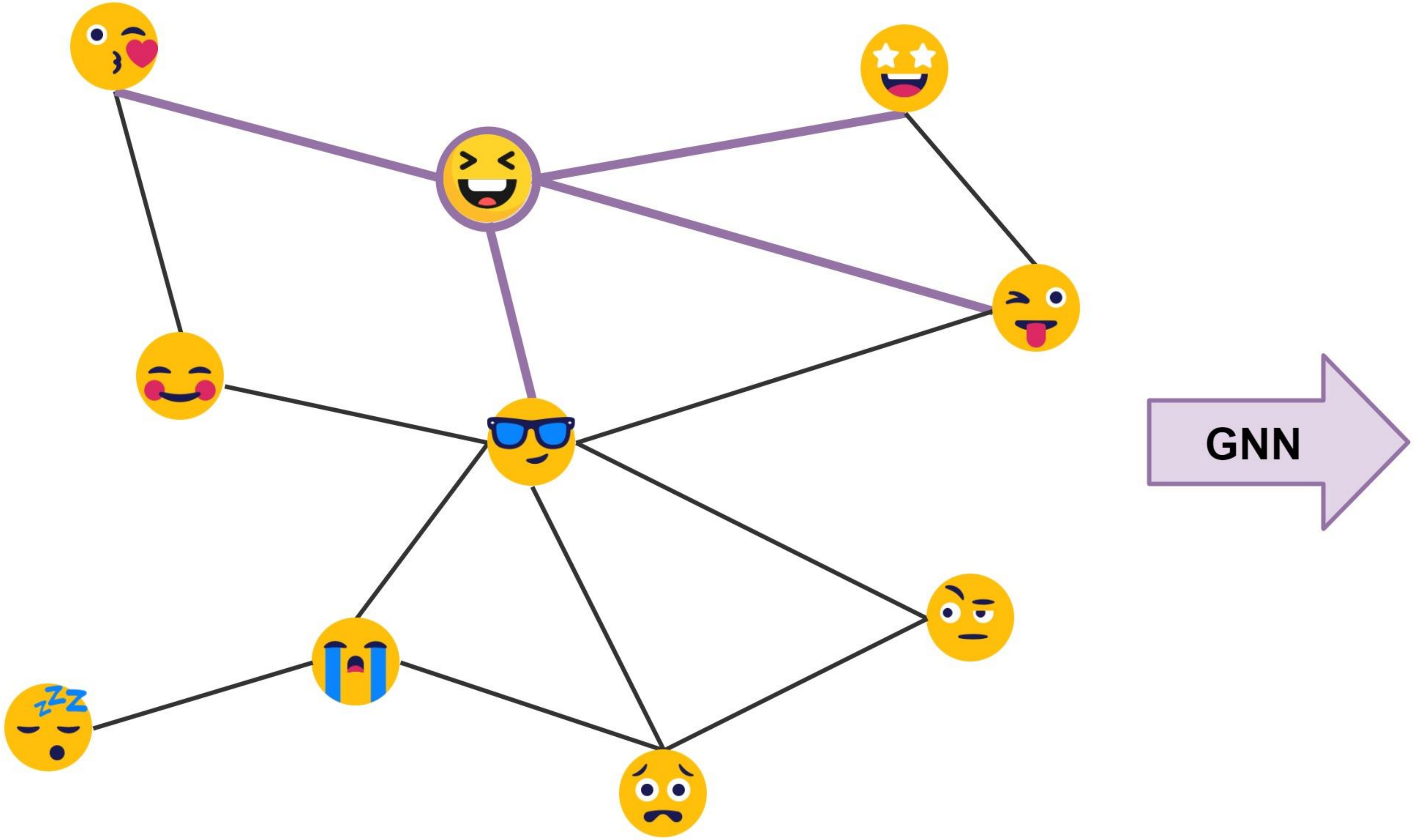
+ Scaling dot product attention

+ Normalization layers

+ Residual links



Representation Learning on Graphs



Most influential?

Least influential?

Possible connection?

Unrelated?

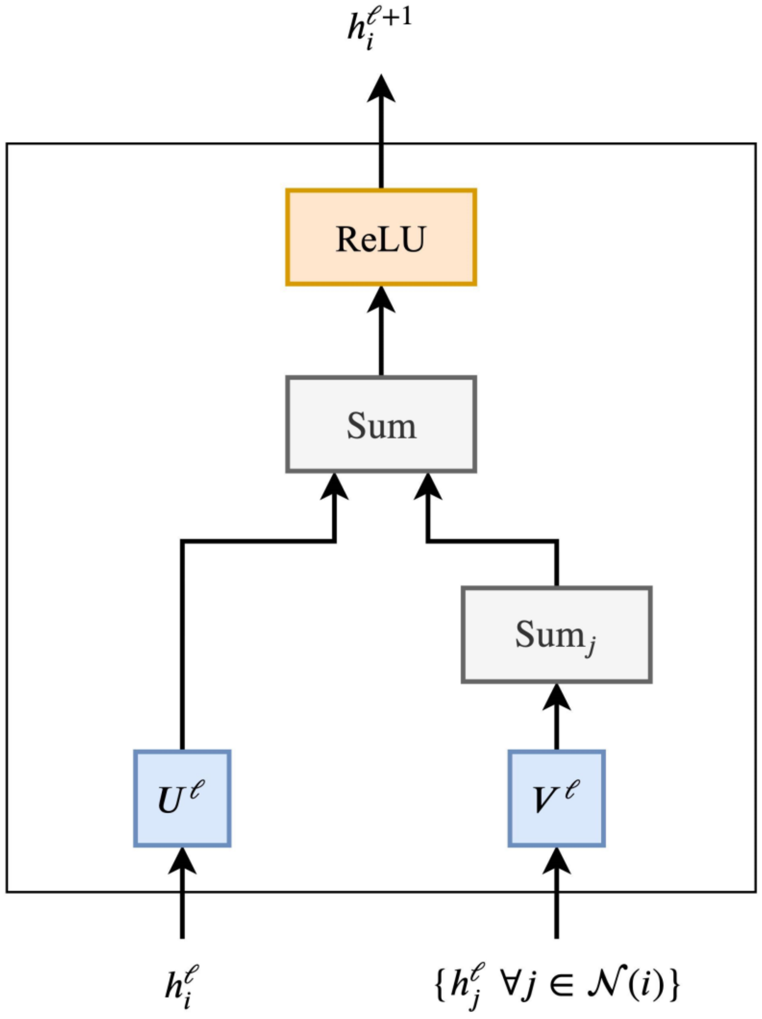
Most similar?

Graph Neural Networks

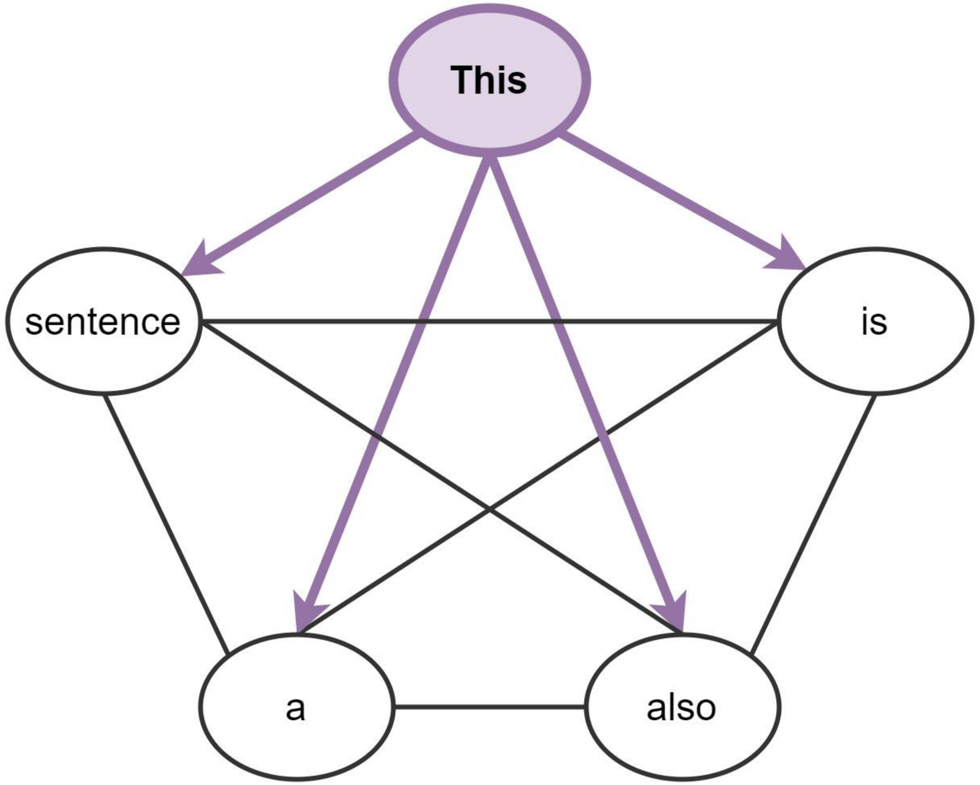
GNNs update the hidden features h of node i at layer ℓ via a non-linear transformation of the node's own features added to the aggregation of features from each neighbouring node $j \in N(i)$:

$$h_i^{\ell+1} = \sigma\Big(U^\ell h_i^\ell + \sum_{j \in \mathcal{N}(i)} \left(V^\ell h_j^\ell \right) \Big),$$

where U, V are learnable weight matrices of the GNN layer and σ is a non-linearity.

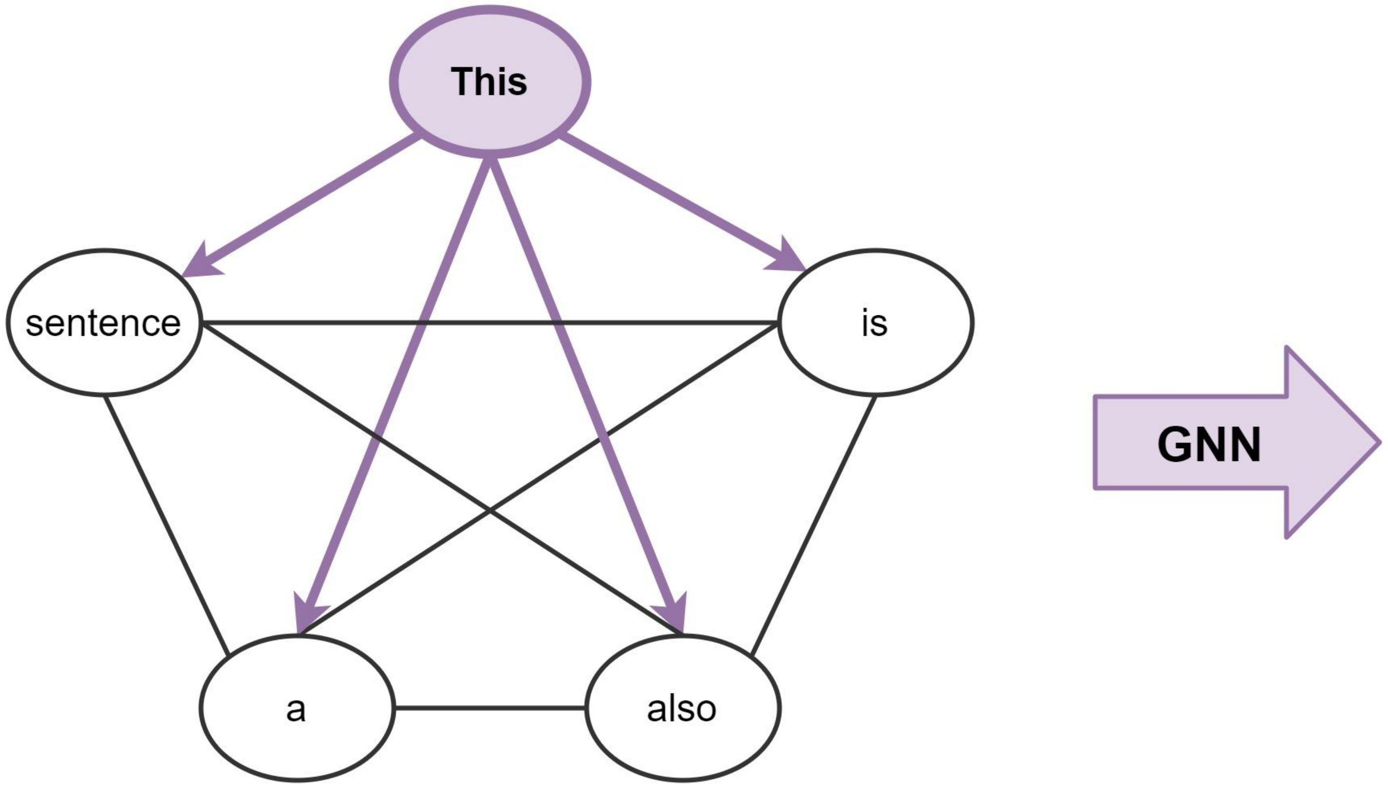


Connections b/w GNNs and Transformers



Consider a sentence as a fully connected graph of words…

Connections b/w GNNs and Transformers

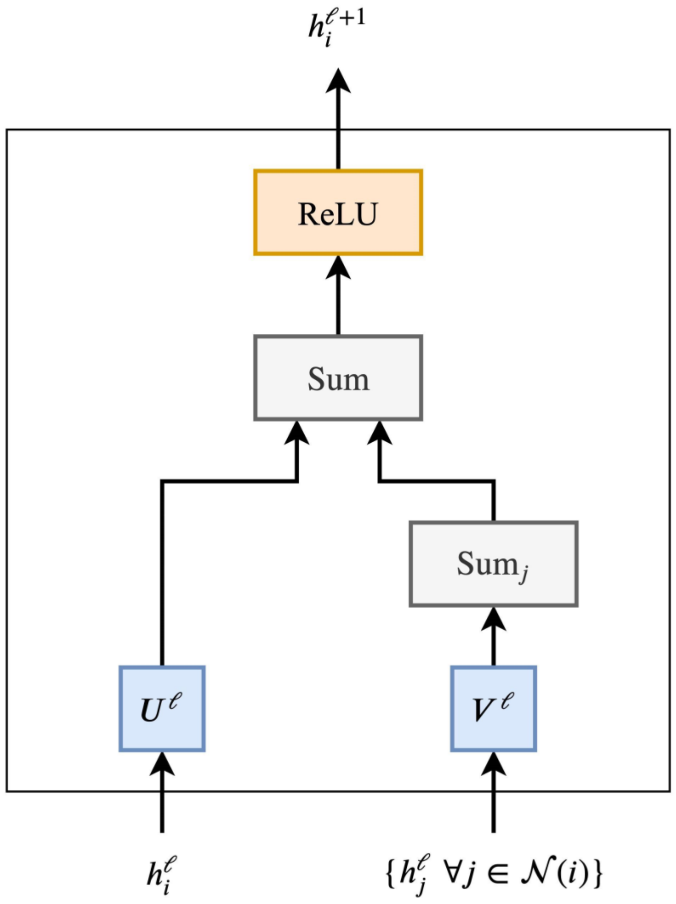


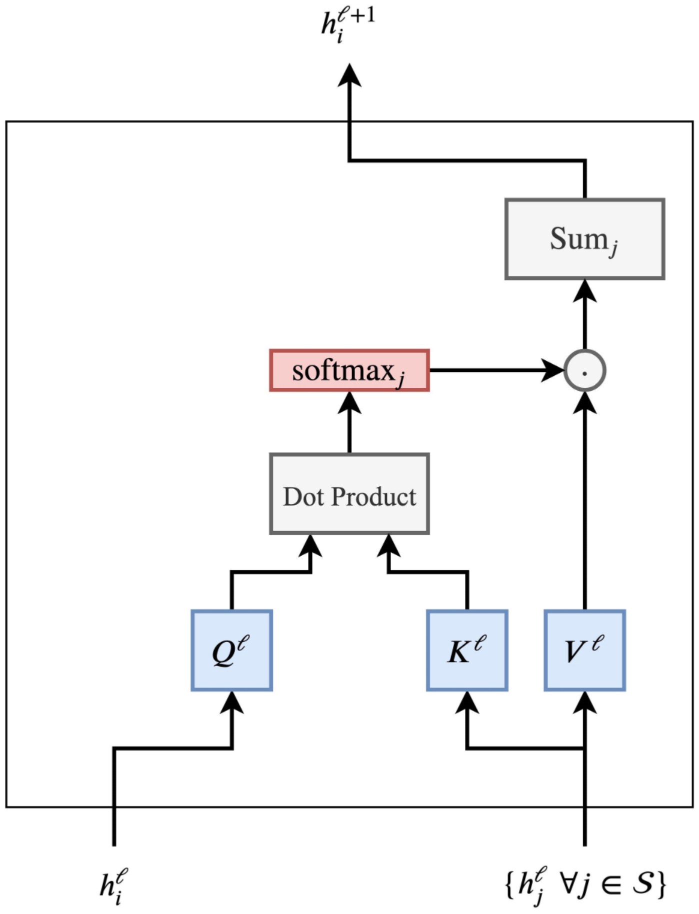
Translation?

Sentiment?

Next word?

Part-of-speech tags?

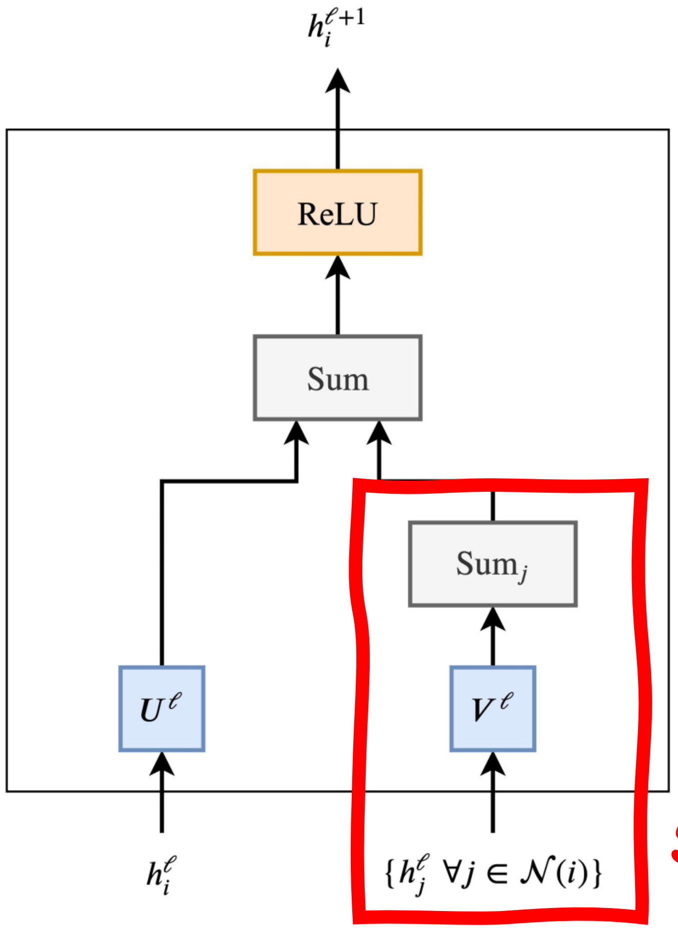




$$h_i^{\ell+1} = \sigma\Big(U^\ell h_i^\ell + \sum_{j \in \mathcal{N}(i)} \left(V^\ell h_j^\ell\right)\Big),$$

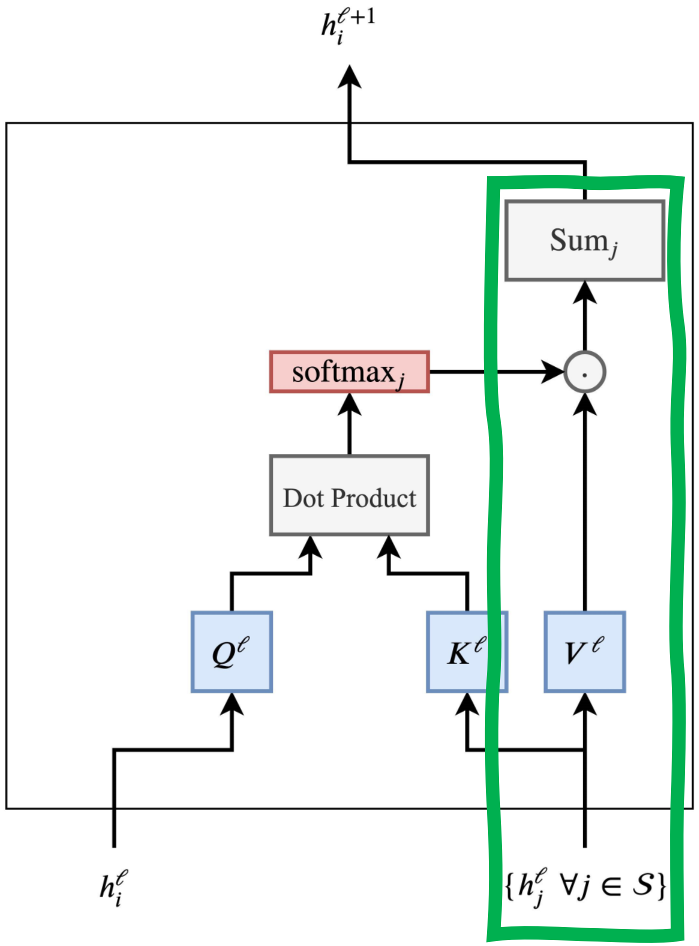
$$i.\,e.,\ h_i^{\ell+1} = \sum_{j \in \mathcal{S}} w_{ij} \left(V^\ell h_j^\ell\right),$$

$$\text{where } w_{ij} = \text{softmax}_j\left(Q^\ell h_i^\ell \cdot K^\ell h_j^\ell\right),$$



Sum over local neighborhood

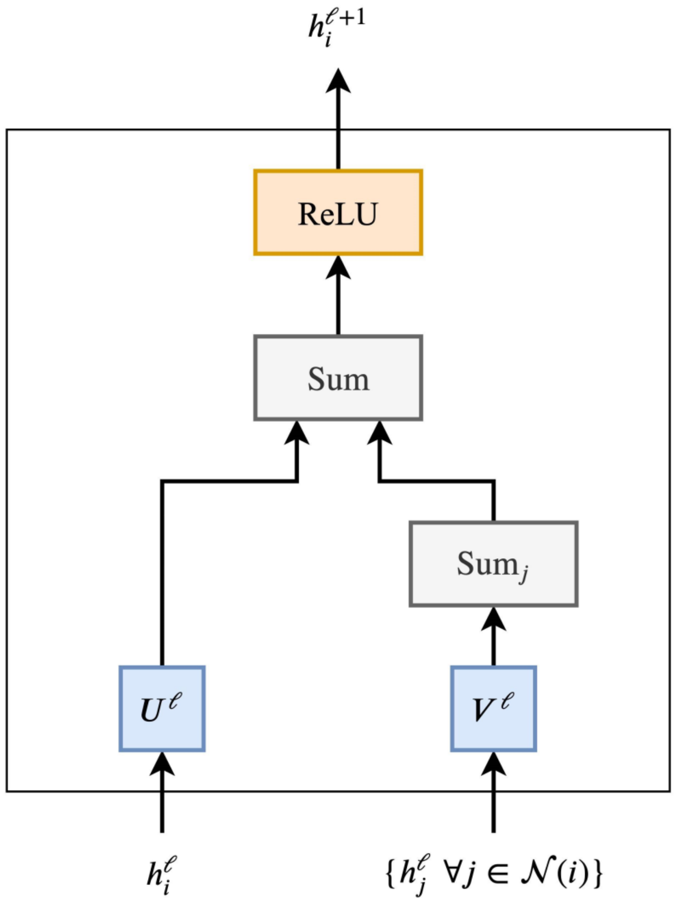
$$h_i^{\ell+1} = \sigma\Big(U^\ell h_i^\ell + \boxed{\sum_{j\in\mathcal{N}(i)} \big(V^\ell h_j^\ell\big)}\Big),$$

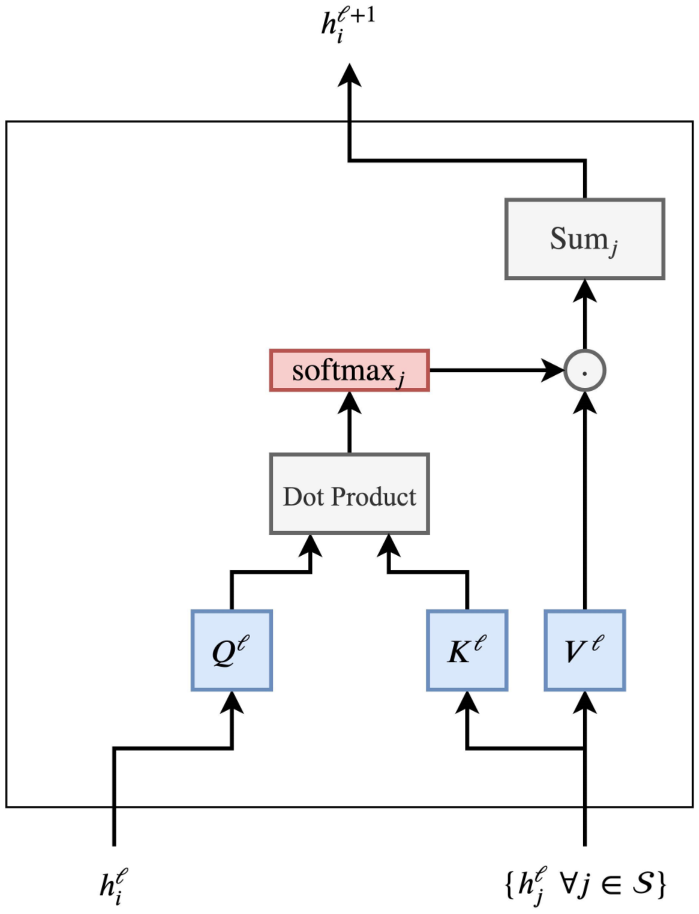


Sum over all words in S

$$i.\,e.,\ h_i^{\ell+1} = \boxed{\sum_{j\in\mathcal{S}} w_{ij}\big(V^\ell h_j^\ell\big),}$$

$$\text{where } w_{ij} = \text{softmax}_j\big(Q^\ell h_i^\ell \cdot K^\ell h_j^\ell\big),$$

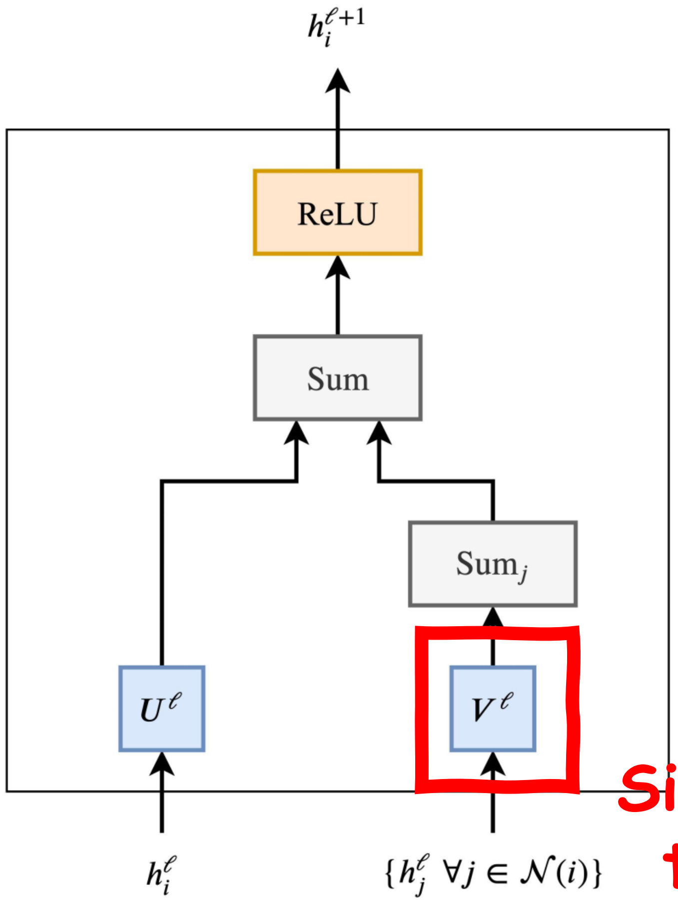




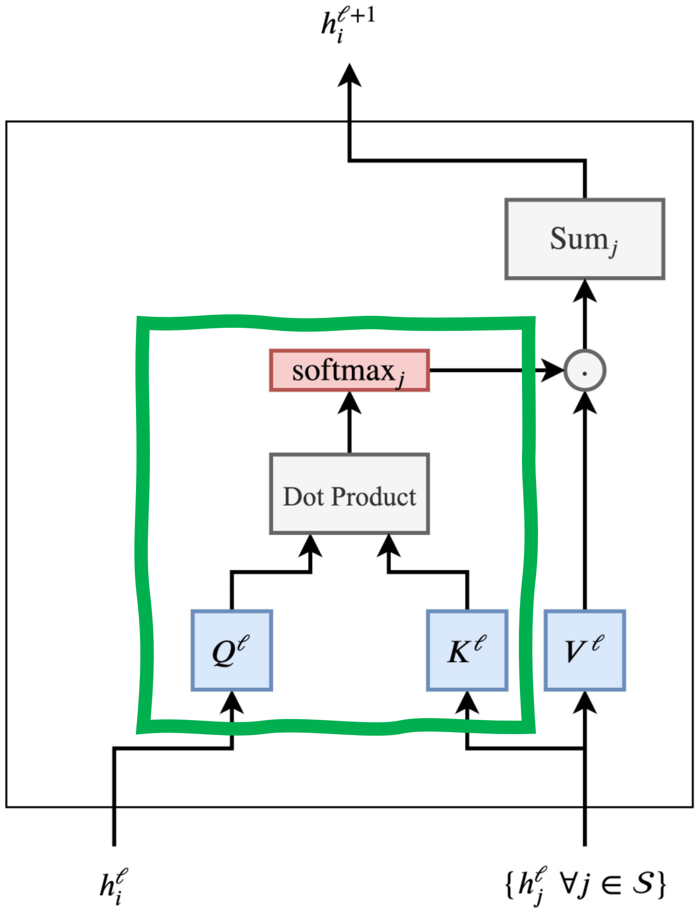
$$h_i^{\ell+1} = \sigma\Big(U^\ell h_i^\ell + \sum_{j\in\mathcal{N}(i)} \left(V^\ell h_j^\ell\right)\Big),$$

$$i.e.,\ h_i^{\ell+1} = \sum_{j\in\mathcal{S}} w_{ij}\left(V^\ell h_j^\ell\right),$$

$$\text{where } w_{ij} = \text{softmax}_j\left(Q^\ell h_i^\ell \cdot K^\ell h_j^\ell\right),$$



Simple linear transform and sum

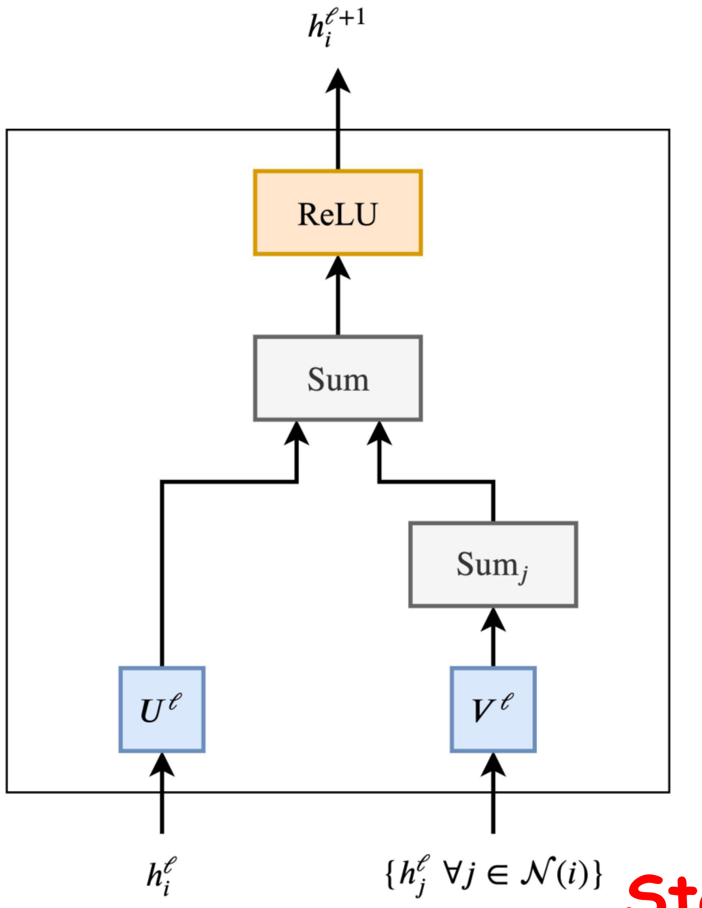


Weighted sum via attention

$$h_i^{\ell+1} = \sigma\Big(U^\ell h_i^\ell + \sum_{j\in\mathcal{N}(i)} \boxed{(V^\ell h_j^\ell)}\Big),$$

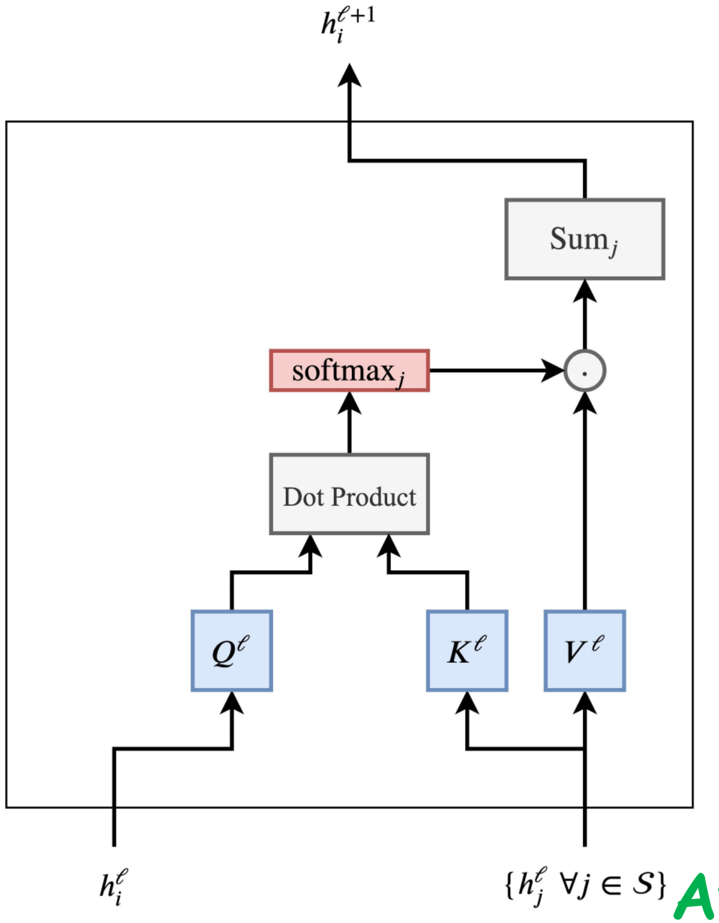
$$i.e., \ h_i^{\ell+1} = \sum_{j\in\mathcal{S}} \boxed{w_{ij}}\,(V^\ell h_j^\ell),$$

where $\boxed{w_{ij} = \mathrm{softmax}_j(Q^\ell h_i^\ell \cdot K^\ell h_j^\ell),}$



Standard Graph NN

$$h_i^{\ell+1} = \sigma\Big(U^\ell h_i^\ell + \sum_{j \in \mathcal{N}(i)} \big(V^\ell h_j^\ell\big)\Big),$$

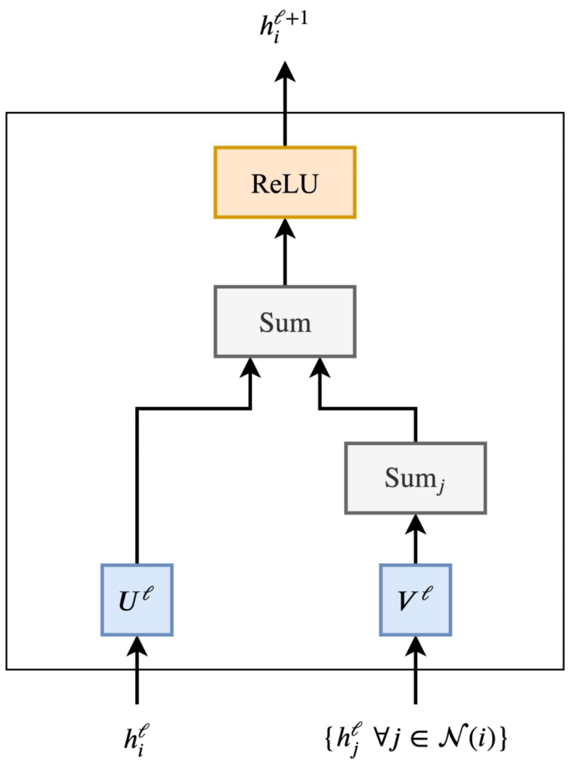


Graph Attention Network

$$i.\,e.,\ h_i^{\ell+1} = \sum_{j \in \mathcal{S}} w_{ij}\big(V^\ell h_j^\ell\big),$$

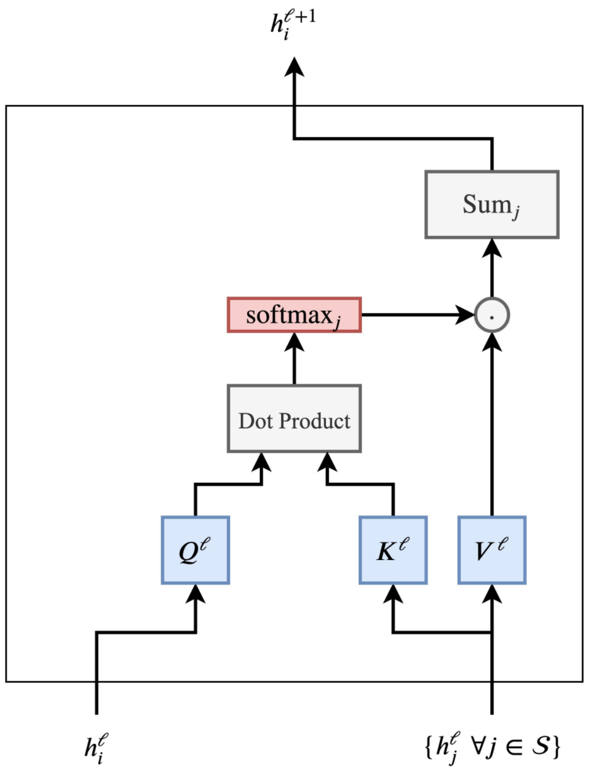
$$\text{where } w_{ij} = \mathrm{softmax}_j\big(Q^\ell h_i^\ell \cdot K^\ell h_j^\ell\big),$$

Standard GNN



+ Weighted sum aggregation

GAT



+ Multi-head mechanism

+ Normalization layers

+ Residual links

Transformer