Attention Is All You Need

Ashish Vaswani, Noam Shazeer, Niki Parmar

arXiv:1706.03762, NeurIPS 2017

SuYong Jeong
Data Mining And Intelligence System LAB

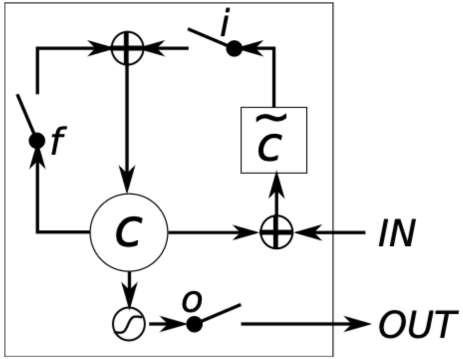
Outline

- ☐ Sequence Modeling
- ☐ Transformer
- ☐ Experiments
- ☐ Conclusion

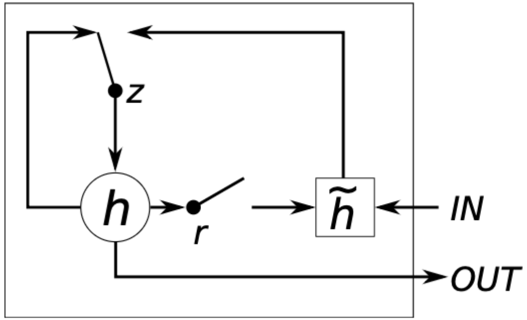
Sequence Modeling

☐ What is Sequence Modeling?

- Sequence modeling refers to the comprehensive modeling of sequential data
  - ☐ Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU) are dominant models for processing sequential data



(a) Long Short-Term Memory

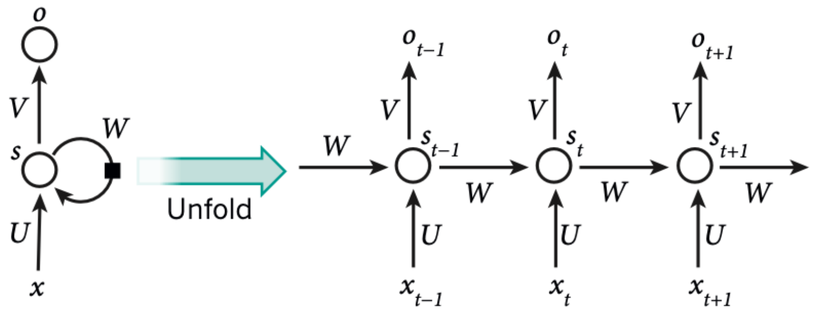


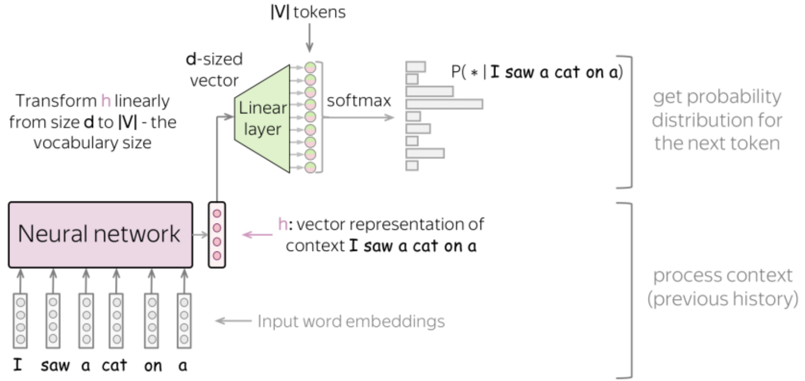
(b) Gated Recurrent Unit

Sequence Modeling

Language Modeling with RNN

- RNN processes input sequences one step at a time, **accumulating information** in the hidden state
- The final hidden state of an RNN contains the representation of the entire sentence

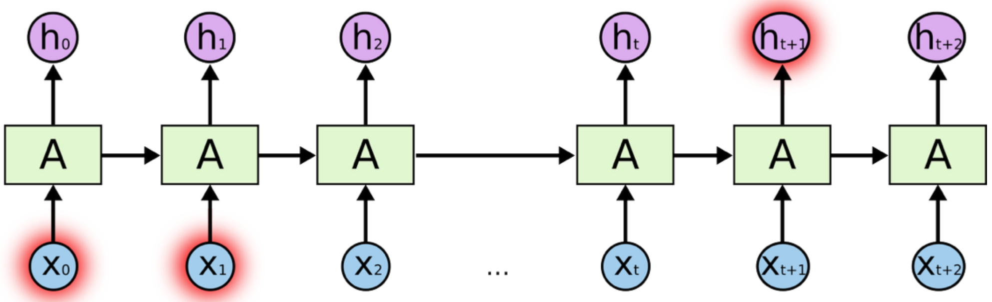




Sequence Modeling

☐ Challenge of Conventional Sequence Modeling

- As the sentence gets longer, RNN becomes **less capable of learning long-range dependencies** between distant words
- The nature of sequential modeling **prevents parallelization** within a training example
  - It becomes critical as the sequence length increases because memory constraints limit batching



Sequence Modeling

☐ Using Convolutional Neural Networks

- Treating sentences with CNNs can compute hidden representations in parallel for all input and output positions
  - ☐ But still, this makes it difficult to learn dependencies between distant positions

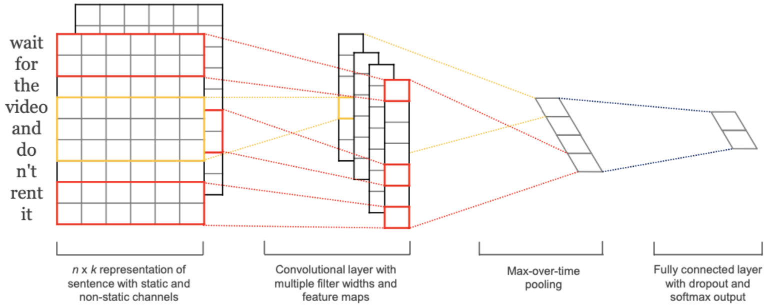
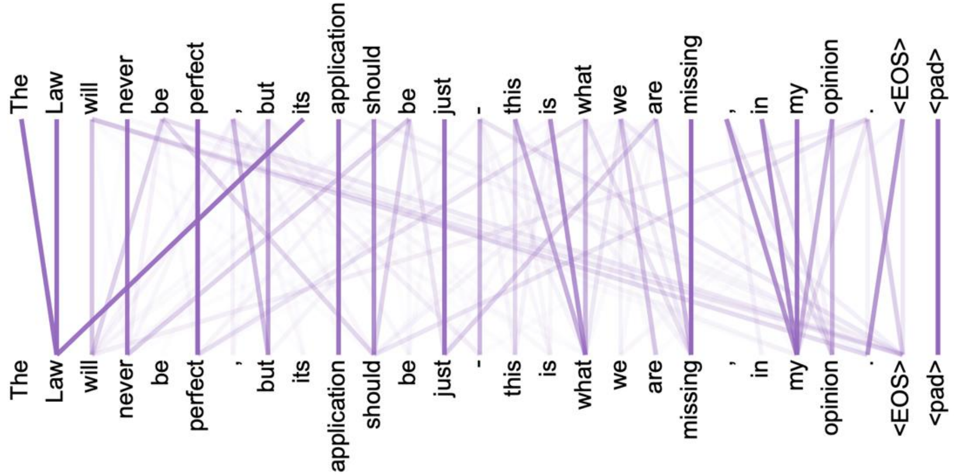


Figure 1: Model architecture with two channels for an example sentence.

Transformer

☐ Self-Attention Mechanism

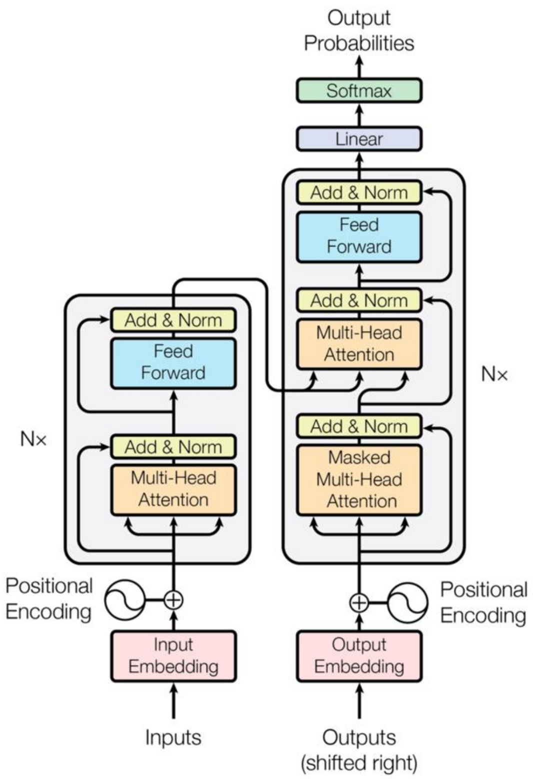
- Self-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence
  - ☐ It enables the model to more effectively capture long-range dependencies than a recurrent model



Transformer

☐ What is Transformer?

- Transformer is the transduction model that **relies entirely on self-attention mechanism** to process sequential data, **without using recurrence or hidden states**

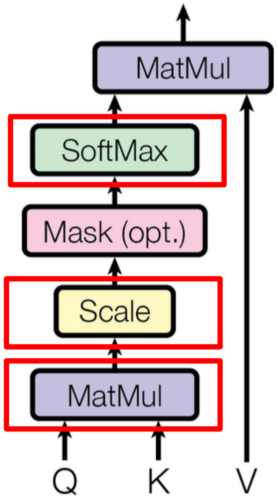


Self-Attention

☐ Scaled Dot-Product Attention

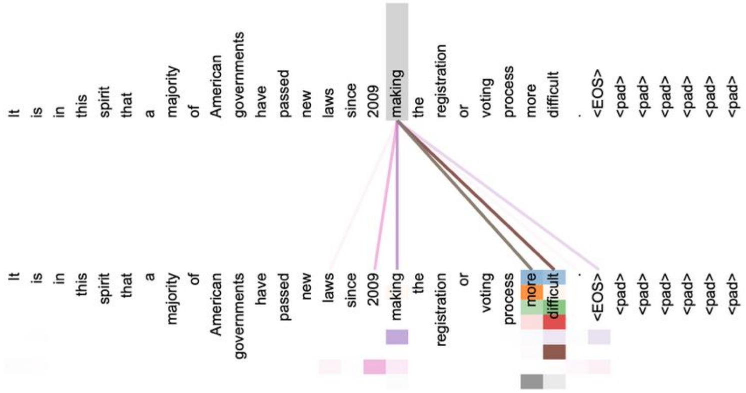
- $\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$
- $Q_i = X_i \times W_Q$, $K_i = X_i \times W_K$, $V_i = X_i \times W_V$
  - Query represents the target word we want to focus on
  - Keys represent all the words in the sentence
  - Values represent the actual meaning or information carried by each Key
- QK^T means **the similarity between the current Query and all Keys** through a dot product

Scaled Dot-Product Attention

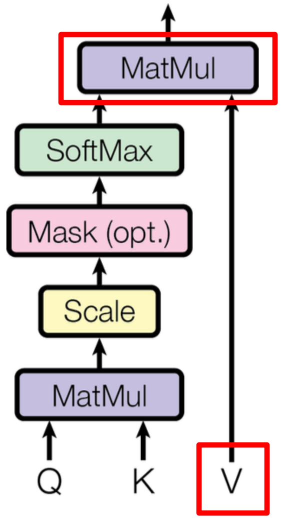


Self-Attention

- ## Scaled Dot-Product Attention
  - $\mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$ means **aggregating the actual information from other words (Key)**, weighted by the attention scores
    - Repeating this process for each word yields a context vector per word, incorporating information from the whole sentence



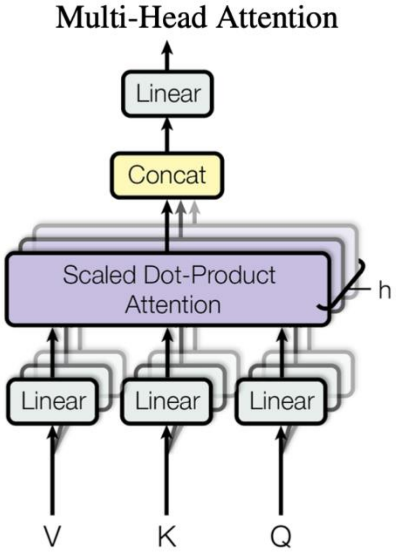
Scaled Dot-Product Attention



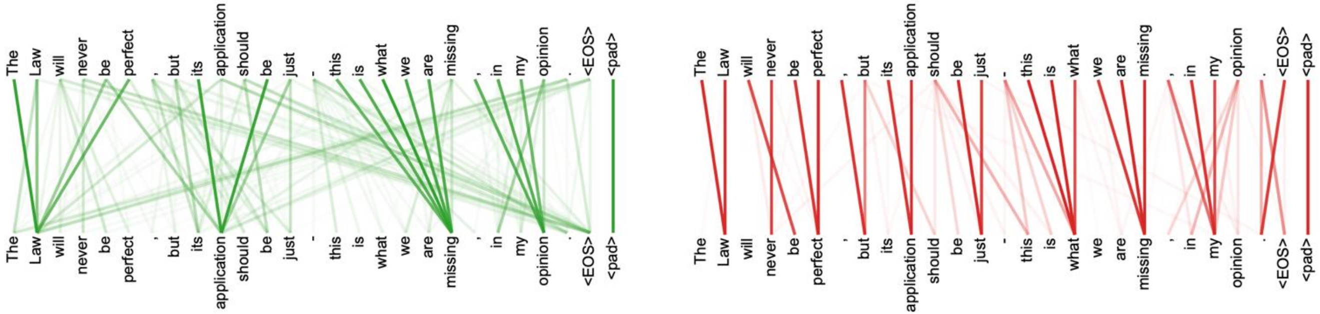
Self-Attention

☐ Multi-Head Attention

- Q, K, and V are computed through linear projections and then **split into smaller dimensions** across multiple heads, where each head performs self-attention independently
  - ☐ Concatenating the outputs of multiple heads produces more generalized representations and improves accuracy



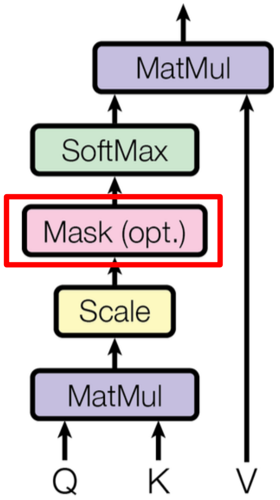
$$\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^O$$
$$\text{where } \mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



Self-Attention

- Masked Multi-Head Attention
  - The mask in the decoder prevents access to future tokens by setting their attention scores to -inf
  - The output is used as the Query to perform attention over the Encoder's output (as the Key and Value)

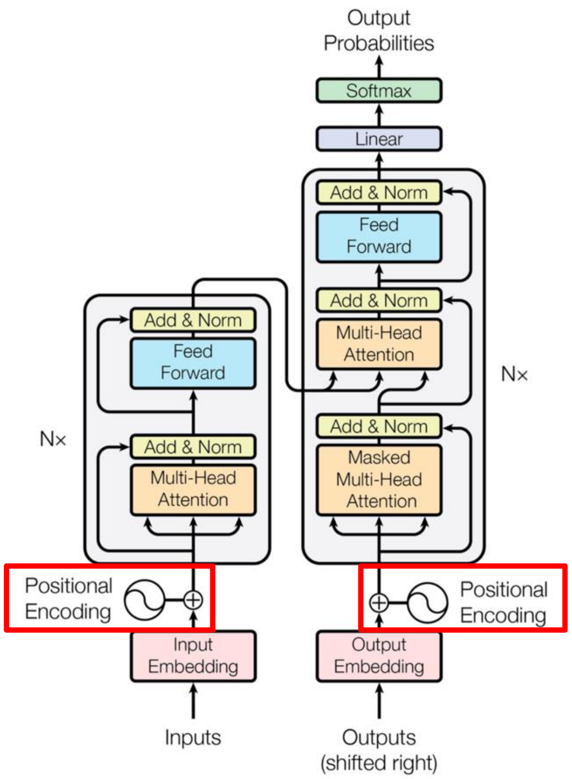
Scaled Dot-Product Attention



Other Components

- **Positional Encoding**
  - Since self-attention operates in parallel without sequence information, positional information must be added for the model to understand token order

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

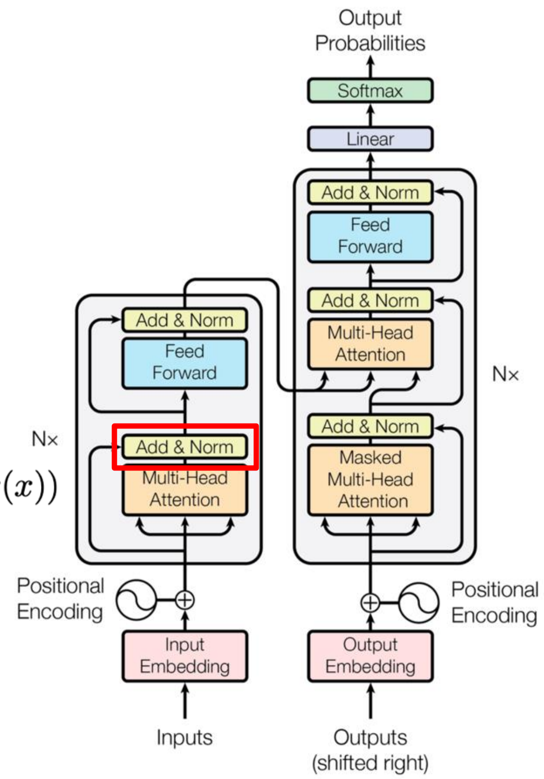


Other Components

☐ Residual Connection

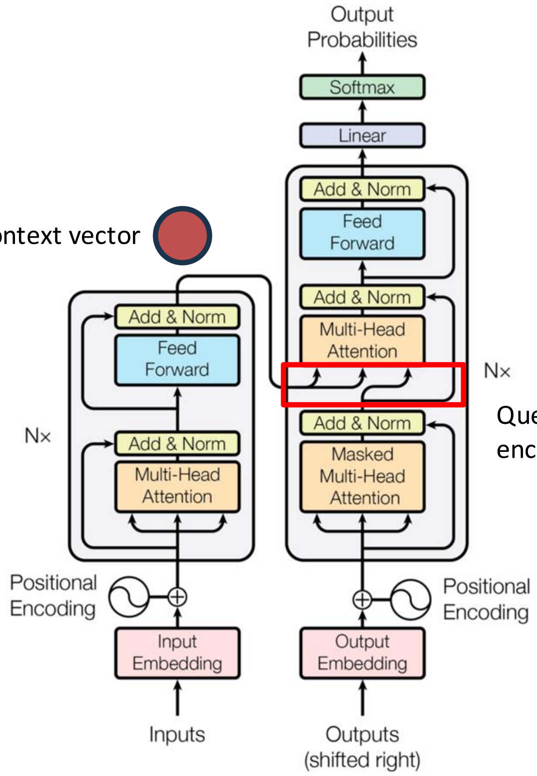
- Residual connection helps preserve input information and improves gradient flow, reducing the risk of vanishing gradients in deep networks

$$\mathrm{LayerNorm}(x + \mathrm{Sublayer}(x))$$



Overall Architecture

Context vector



Query for
encoder's output

Another Example of Transformer

☐ News Recommend System with Attention

- Q = Information of reader

수용 정
Joined: Apr 14, 2023

Member Statistics
Total Analyses: 0
Total News Articles: 0
Total Comments: 0

Information Sentiments

Indices

수용 정 Indices Sentiments

Start Date	Name	Call	Open Rate	End Date	Chg. %
06-01-25	Nasdaq 100		21494.0	10-01-25 @ 20739.0	+3.51%
30-08-24	Nasdaq 100		19430.6	30-08-24 @ 19431.6	-0.01%
11-07-24	Nasdaq 100		20660.8	12-07-24 @ 20421.9	-1.16%
07-06-24	Nasdaq 100		19048.0	14-06-24 @ 19613.8	+2.97%
07-06-24	KOSPI		2718.79	07-06-24 @ 2717.05	+0.06%

- K = Title of news

Stock market today: Nasdaq notches closing record as Nvidia hits $4T

Investing.com | Author Yasin Ebrahim | Stock Markets
Published 07/08/2025, 08:13 PM | Updated 07/09/2025, 04:22 PM

>

China's ETF market is booming, but what comes next?

Investing.com | Author Yasin Ebrahim | Stock Markets
Published 07/09/2025, 05:25 PM

- V = Contents of news

Investing.com-- The Nasdaq closed at record highs Wednesday, shrugging off President Donald Trump's latest tariff blitz as Nvidia rallied to top $4T in value for the first time, pushing the broader tech sector higher.

At 4:00 pm ET (20:00 GMT), the NASDAQ Composite 100 Futures}} climbed 0.95% to a closing record of 20,611.34, the Dow Jones Industrial Average gained 217 points, or 0.5%, and the S&P 500 index rose 0.6%.

Investing.com -- China's exchange traded fund, or ETF, market has exploded in recent years, with passive funds now outmuscling their active rivals and reshaping the landscape of the world's second-largest equity market. But as the ETF boom accelerates, questions are swirling about what this means for A-share liquidity, volatility, and the future of active management.

Experiment

☐ English-to-German, English-to-French

- Transformer model outperforms the best previously reported models (including ensembles) by more than 2.0 BLEU, establishing a new state-of-the-art BLEU score of 28.4.

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	$\mathbf{3.3 \cdot 10^{18}}$	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

Experiment

☐ **Variations on the Transformer architecture**

	N	d_{model}	d_{ff}	h	d_k	d_v	P_{drop}	ϵ_{ls}	train steps	PPL (dev)	BLEU (dev)	params $\times 10^6$
base	6	512	2048	8	64	64	0.1	0.1	100K	4.92	25.8	65
(A)				1	512	512				5.29	24.9	
				4	128	128				5.00	25.5	
				16	32	32				4.91	25.8	
				32	16	16				5.01	25.4	
(B)					16					5.16	25.1	58
					32					5.01	25.4	60
(C)	2									6.11	23.7	36
	4									5.19	25.3	50
	8									4.88	25.5	80
		256			32	32				5.75	24.5	28
		1024			128	128				4.66	26.0	168
			1024							5.12	25.4	53
			4096							4.75	26.2	90
(D)							0.0			5.77	24.6	
							0.2			4.95	25.5	
								0.0		4.67	25.3	
								0.2		5.47	25.7	
(E)	positional embedding instead of sinusoids									4.92	25.7	
big	6	1024	4096	16			0.3		300K	**4.33**	**26.4**	213

Experiment

□ English Constituency Parsing

- The Transformer performs well not only in word prediction but also in more complex tasks such as constituency parsing

Table 4: The Transformer generalizes well to English constituency parsing (Results are on Section 23 of WSJ)

Parser	Training	WSJ 23 F1
Vinyals & Kaiser el al. (2014) [37]	WSJ only, discriminative	88.3
Petrov et al. (2006) [29]	WSJ only, discriminative	90.4
Zhu et al. (2013) [40]	WSJ only, discriminative	90.4
Dyer et al. (2016) [8]	WSJ only, discriminative	91.7
Transformer (4 layers)	WSJ only, discriminative	91.3
Zhu et al. (2013) [40]	semi-supervised	91.3
Huang & Harper (2009) [14]	semi-supervised	91.3
McClosky et al. (2006) [26]	semi-supervised	92.1
Vinyals & Kaiser el al. (2014) [37]	semi-supervised	92.1
Transformer (4 layers)	semi-supervised	92.7
Luong et al. (2015) [23]	multi-task	93.0
Dyer et al. (2016) [8]	generative	93.3

Conclusion

- ☐ The **Transformer** is a sequential modeling architecture that **based solely on self-attention without using sequential recurrence**

- ☐ This allows it to **learn relationships between tokens** regardless of their distance, while enabling parallel computation for faster processing

- ☐ The model **generalizes well to more complex tasks** beyond simple word prediction

Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks

Juho Lee, Yoonho Lee, Jungtaek Kim

ICML 2019

SuYong Jeong
Data Mining And Intelligence System LAB

Outline

- ☐ Set-Input Problem
- ☐ Set Pooling
- ☐ Set Transformer
- ☐ Experiments
- ☐ Conclusion

Set-Input Problem

☐ What is Set-Input problem?

- Given a set of instances as an input, the corresponding target is a label for the entire set
  - ☐ Multiple instance learning
  - ☐ 3D shape recognition

Serge's key-chain

Serge **cannot** enter the *Secret Room*

Sanjoy's key-chain

Sanjoy **can** enter the *Secret Room*

Lawrence's key-chain

Lawrence **can** enter the *Secret Room*

Set-Input Problem

☐ Critical Requirements for Set-Input Problems

- Stem from the definition of a set
  - ☐ It should be permutation invariant
  - ☐ A model should be able to process input sets of any size

They are not easily satisfied in neural-network-based models!

Set Pooling

What is Set Pooling method?

- Set Pooling is neural network architectures which meet both criteria
  - Each element in a set is independently fed into a feed-forward neural network
  - Resulting feature-space embeddings are aggregated using a pooling operation
- A permutation-equivariant encoder followed by a permutation-invariant decoder yields a permutation-invariant network

$$\mathrm{net}(\{x_1, \ldots, x_n\}) = \rho(\mathrm{pool}(\{\phi(x_1), \ldots, \phi(x_n)\}))$$

Set Pooling

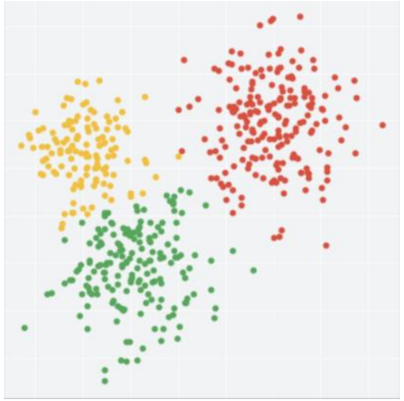
- What is Set Pooling method?
  - Every element in a set is processed independently in a set pooling operation
    - Some information regarding interactions between elements has to be necessarily discarded
    - This can make some problems unnecessarily difficult to solve

$$\mathrm{net}(\{x_1, \ldots, x_n\}) = \rho(\boxed{\mathrm{pool}}(\{\boxed{\phi(x_1)}, \ldots, \phi(x_n)\}))$$

Set Pooling

☐ Amortized Clustering

- Learning a parametric mapping from an input set of points to the centers of clusters
  - The parametric mapping must assign each point to its corresponding cluster while modelling the explaining away pattern
- Set pooling don't care interactions between elements, making it unsuitable for modeling relationships or enabling explaining away among cluster centers



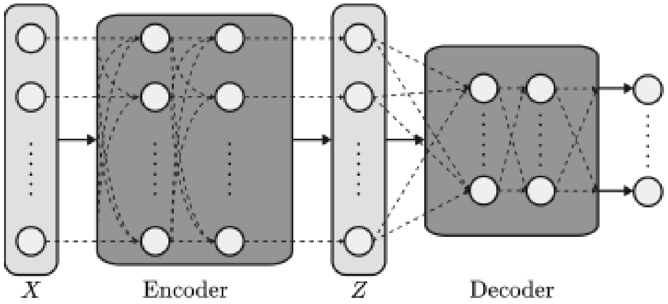
Set Transformer

- Self-attention
  - Using Self-attention, we can measure how similar each pair of query and key vectors is
    - The output $\omega(QK^{\top})V$ is a weighted sum of V where a value gets more weight if its corresponding key has larger dot product with the query
    - We can encode the interactions between elements in the set with self-attention

Set Transformer

☐ What is Set Transformer?

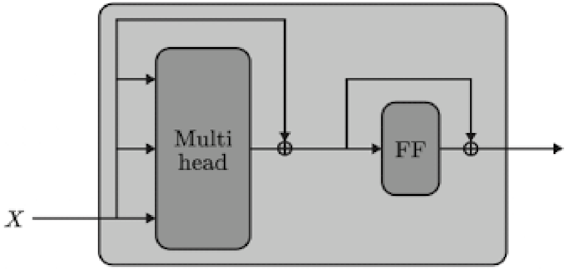
- Set Transformer is an attention-based neural network that is designed to process sets of data
  - ☐ It consists of an encoder followed by a decoder
  - ☐ Each layer in the encoder and decoder attends to their inputs to produce activations
  - ☐ Its aggregating function is parameterized and can adapt to the problem at hand



Set Transformer

Set Attention Block

- Set Transformer uses self-attention to concurrently encode the whole set
  - This gives the Set Transformer the ability to compute pairwise as well as higher-order interactions among instances during the encoding process
  - A potential problem with using SABs for set-structured data is the quadratic time complexity $\mathrm{O}(\mathrm{n}^2)$



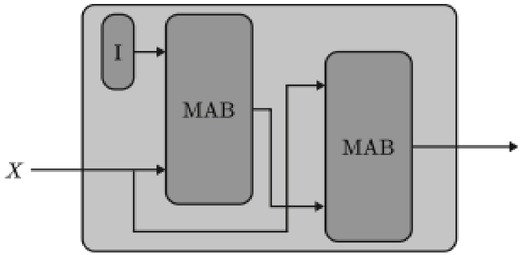
(c) SAB

$$\mathrm{MAB}(X, Y) = \mathrm{LayerNorm}(H + \mathrm{rFF}(H)),$$
$$\text{where } \; H = \mathrm{LayerNorm}(X + \mathrm{Multihead}(X, Y, Y; \omega))$$
$$\mathrm{SAB}(X) := \mathrm{MAB}(X, X).$$

Set Transformer

Induced Set Attention Block

- Along with the set $X \in \mathbb{R}^{n \times d}$, additionally define m d-dimensional vectors $I \in \mathbb{R}^{m \times d}$, which we call inducing points
  - Inducing points, I are part of the ISAB itself, and they are trainable parameters
  - The learned inducing points are expected to encode some global structure which helps explain the inputs X
  - The time complexity of $\mathrm{ISAB_m}(\mathrm{X}; \lambda)$ is $\mathrm{O(nm)}$ where m is a hyperparameter



(d) ISAB

$$\mathrm{ISAB}_m(X) = \mathrm{MAB}(X, H) \in \mathbb{R}^{n \times d},$$
$$\text{where } H = \mathrm{MAB}(I, X) \in \mathbb{R}^{m \times d}.$$

Set Transformer

Pooling by Multihead Attention

- Features from encoder are aggregated by applying multihead attention on a learnable set of k seed vectors $S \in \mathbb{R}^{k \times d}$
  - We use one seed vector (k= 1) in most cases, but for problems such as amortized clustering which requires k correlated outputs, the natural thing to do is to use k seed vectors

$$\mathrm{PMA}_k(Z) = \mathrm{MAB}(S, \mathrm{rFF}(Z)).$$

Set Transformer

☐ Overall Architecture

- Encoder: $X \mapsto Z \in \mathbb{R}^{n \times d}$ is a stack of SABs or ISABs
  - ☐ $\mathrm{Encoder}(X) = \mathrm{SAB}(\mathrm{SAB}(X))$
    $\mathrm{Encoder}(X) = \mathrm{ISAB}_m(\mathrm{ISAB}_m(X)).$
  - ☐ After the encoder transforms data $X \in \mathbb{R}^{n \times d_x}$ into features $Z \in \mathbb{R}^{n \times d}$
- Decoder: Aggregating Z into a single or a set of vectors to get final outputs
  - ☐ $\mathrm{Decoder}(Z; \lambda) = \mathrm{rFF}(\mathrm{SAB}(\mathrm{PMA}_k(Z))) \in \mathbb{R}^{k \times d}$
    where $\mathrm{PMA}_k(Z) = \mathrm{MAB}(S, \mathrm{rFF}(Z)) \in \mathbb{R}^{k \times d},$

Experiments

☐ Maximum Value Regression

Table 1. Mean absolute errors on the max regression task.

Architecture	MAE
rFF + Pooling (mean)	2.133 ± 0.190
rFF + Pooling (sum)	1.902 ± 0.137
rFF + Pooling (max)	$\mathbf{0.1355 \pm 0.0074}$
SAB + PMA (ours)	0.2085 ± 0.0127

Experiments

☐ Counting Unique Characters

Table 2. Accuracy on the unique character counting task.

Architecture	Accuracy
rFF + Pooling	0.4382 ± 0.0072
rFFp-mean + Pooling	0.4617 ± 0.0076
rFFp-max + Pooling	0.4359 ± 0.0077
rFF + Dotprod	0.4471 ± 0.0076
rFF + PMA (ours)	0.4572 ± 0.0076
SAB + Pooling (ours)	0.5659 ± 0.0077
SAB + PMA (ours)	$\mathbf{0.6037 \pm 0.0075}$

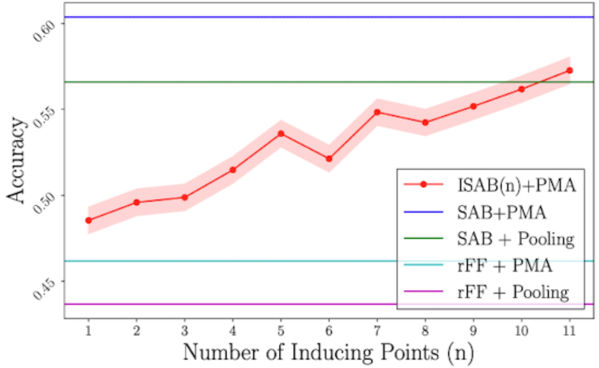


Figure 3. Accuracy of ISAB_n + PMA on the unique character counting task. x-axis is n and y-axis is accuracy.

Experiments

☐ Amortized Clustering

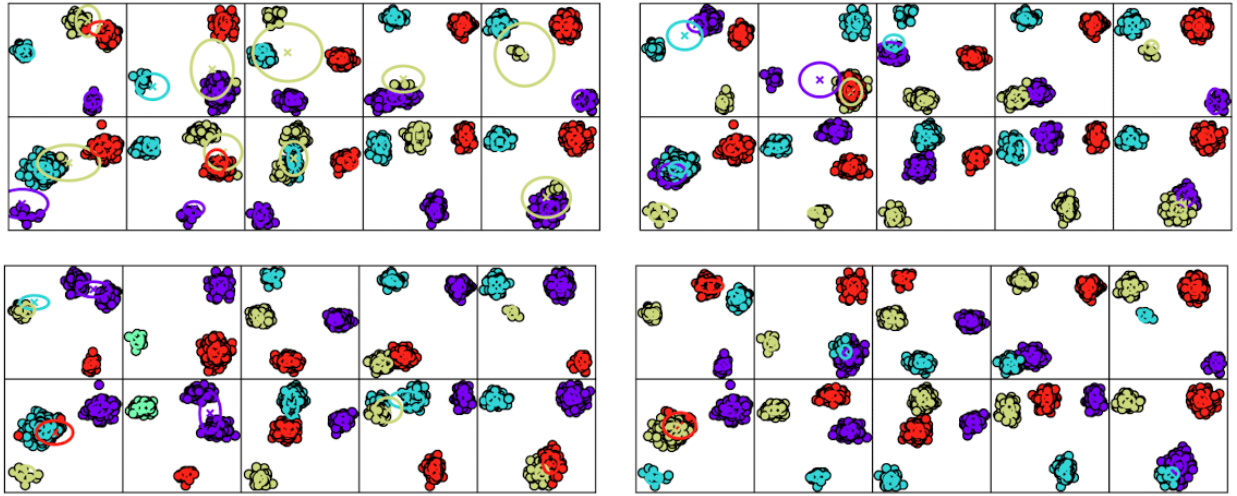


Figure 4. Clustering results for 10 test datasets, along with centers and covariance matrices. rFF+Pooling (top-left), SAB+Pooling (top-right), rFF+PMA (bottom-left), Set Transformer (bottom-right). Best viewed magnified in color.

Experiments

☐ Set Anomaly Detection

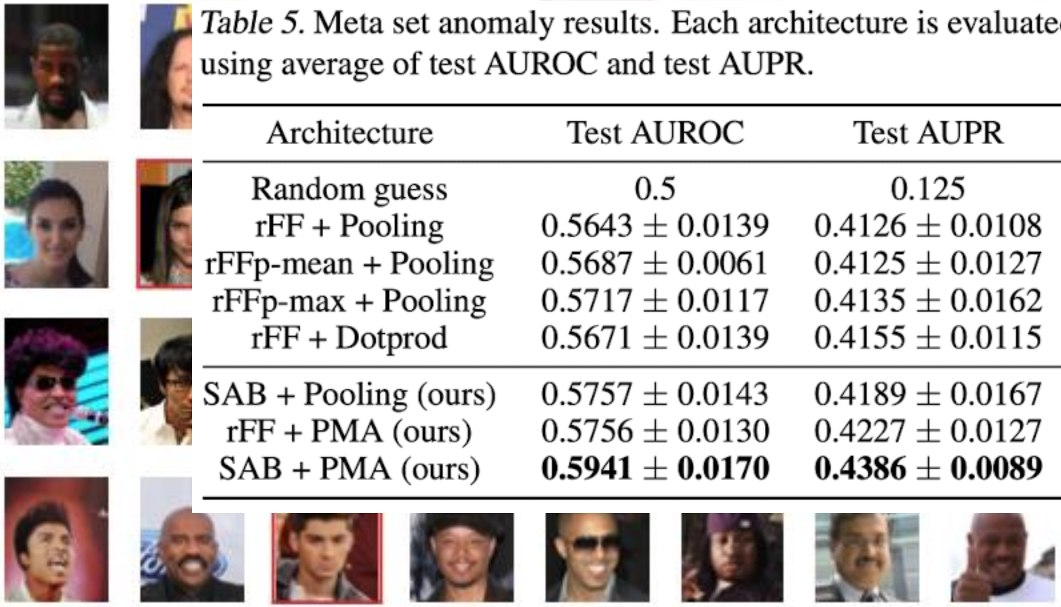


Table 5. Meta set anomaly results. Each architecture is evaluated using average of test AUROC and test AUPR.

Architecture	Test AUROC	Test AUPR
Random guess	0.5	0.125
rFF + Pooling	0.5643 ± 0.0139	0.4126 ± 0.0108
rFFp-mean + Pooling	0.5687 ± 0.0061	0.4125 ± 0.0127
rFFp-max + Pooling	0.5717 ± 0.0117	0.4135 ± 0.0162
rFF + Dotprod	0.5671 ± 0.0139	0.4155 ± 0.0115
SAB + Pooling (ours)	0.5757 ± 0.0143	0.4189 ± 0.0167
rFF + PMA (ours)	0.5756 ± 0.0130	0.4227 ± 0.0127
SAB + PMA (ours)	$\mathbf{0.5941 \pm 0.0170}$	$\mathbf{0.4386 \pm 0.0089}$

Experiments

☐ Point Cloud Classification

Table 4. Test accuracy for the point cloud classification task using 100, 1000, 5000 points.

Architecture	100 pts	1000 pts	5000 pts
rFF + Pooling (Zaheer et al., 2017)	-	0.83 ± 0.01	-
rFFp-max + Pooling (Zaheer et al., 2017)	0.82 ± 0.02	0.87 ± 0.01	$\mathbf{0.90 \pm 0.003}$
rFF + Pooling	0.7951 ± 0.0166	0.8551 ± 0.0142	0.8933 ± 0.0156
rFF + PMA (ours)	0.8076 ± 0.0160	0.8534 ± 0.0152	0.8628 ± 0.0136
ISAB (16) + Pooling (ours)	0.8273 ± 0.0159	$\mathbf{0.8915 \pm 0.0144}$	$\mathbf{0.9040 \pm 0.0173}$
ISAB (16) + PMA (ours)	$\mathbf{0.8454 \pm 0.0144}$	0.8662 ± 0.0149	0.8779 ± 0.0122

Conclusion

- ☐ The Set Transformer is an attention-based set-input neural network architecture
- ☐ It can model complicated interactions among elements of a set, using attention mechanisms for both encoding and aggregating features
- ☐ With inducing point method for self-attention, it can be scalable to large sets

Thank you