Neurocomputing 452 (2021) 48–62

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A review on the attention mechanism of deep learning

Zhaoyang Niu [a], Guoqiang Zhong [a,*], Hui Yu [b,*]

[a] Department of Computer Science and Technology, Ocean University of China, Qingdao 266100, China
[b] School of Creative Technologies, University of Portsmouth, Portsmouth PO1 2DJ, UK

ARTICLE INFO

Article history:
Received 7 February 2021
Revised 23 March 2021
Accepted 29 March 2021
Available online 1 April 2021
Communicated by Zidong Wang

Keywords:
Attention mechanism
Deep learning
Recurrent Neural Network (RNN)
Convolutional Neural Network (CNN)
Encoder-decoder
Unified attention model
Computer vision applications
Natural language processing applications

ABSTRACT

Attention has arguably become one of the most important concepts in the deep learning field. It is inspired by the biological systems of humans that tend to focus on the distinctive parts when processing large amounts of information. With the development of deep neural networks, attention mechanism has been widely used in diverse application domains. This paper aims to give an overview of the state-of-the-art attention models proposed in recent years. Toward a better general understanding of attention mechanisms, we define a unified model that is suitable for most attention structures. Each step of the attention mechanism implemented in the model is described in detail. Furthermore, we classify existing attention models according to four criteria: the softness of attention, forms of input feature, input representation, and output representation. Besides, we summarize network architectures used in conjunction with the attention mechanism and describe some typical applications of attention mechanism. Finally, we discuss the interpretability that attention brings to deep learning and present its potential future trends.

1. Introduction

Attention is a complex cognitive function that is indispensable for human beings [1,2]. One important property of perception is that humans do not tend to process whole information in its entirety at once. Instead, humans tend to selectively concentrate on a part of the information when and where it is needed, but ignore other perceivable information at the same time. For instance, humans usually don't see all the scenes from the beginning to the end when visually perceiving things, but instead, observe and pay attention to specific parts as needed. When humans find that a scene often has something they want to observe in a certain part, they will learn to focus on that part when similar scenes appear again and focus more attention on the useful part. This is a means for humans to quickly select high-value information from massive information using limited processing resources. The attention mechanism greatly improves the efficiency and accuracy of perceptual information processing.

The attention mechanism of humans can be divided into two categories according to its generation manner [3]. The first category is the bottom-up unconscious attention, called saliency-based attention, which is driven by external stimuli. For example, people are more likely to hear loud voices during a conversation. It is similar to the max-pooling and gating mechanism [4,5] in deep learning, which passes more appropriate values (i.e., larger values) to the next step. The second category is top-down conscious attention, called focused attention. Focused attention refers to the attention that has a predetermined purpose and relies on specific tasks. It enables humans to focus attention on a certain object consciously and actively. Most of the attention mechanisms in deep learning are designed according to specific tasks so that most of them are focused attention. The attention mechanism introduced in this paper usually refers to focused attention except for special statements.

As mentioned above, attention mechanism can be used as a resource allocation scheme, which is the main means to solve the problem of information overload. In the case of limited computing power, it can process more important information with limited computing resources. Hence, some researchers bring attention to the computer vision area. [6] proposed a saliency-based visual attention model that extracts local low-level visual features to get some potential salient regions. In the neural network area, [7] used the attention mechanism on the recurrent neural network model to classify images. Bahdanau et al. [8] used the attention mechanism to simultaneously perform translation and alignment on machine translation tasks. Subsequently, attention mechanism has become an increasingly common ingredient of neural architectures and has been applied to various tasks, such as image caption

* Corresponding author.
E-mail addresses: nnniuzy@163.com (Z. Niu), gqzhong@ouc.edu.cn (G. Zhong), hui.yu@port.ac.uk (H. Yu).

https://doi.org/10.1016/j.neucom.2021.03.091
0925-2312/

generation [9,10], text classification [11,12], machine translation [13–16], action recognition [17–20], image-based analysis [21,22], speech recognition [23–25], recommendation [26,27], and graph [28,29]. Fig. 1 shows an overview of several typical attention methods, which are described in detail in Section 3.

In addition to providing performance improvements, the attention mechanism can also be used as a tool to explain incomprehensible neural architecture behavior. In recent years, neural network has achieved great success in financial [31], material [32], meteorology [33–36], medical [37–41], autonomous driving [42], human–computer interaction [43–45], behavior and action analysis [46,47], industries [48] and emotion detection [49,50], but the lack of interpretability is facing both practical and ethical issues [51]. The interpretability of deep learning has been a problem so far. Although whether the attention mechanism can be used as a reliable method to explain deep networks is still a controversial issue [52,53], it can provide an intuitive explanation to a certain extent [54–57]. For instance, Fig. 2 shows an example of the visualization of attention weight.

This survey is structured as follows. In Section 2, we introduce a well-known model proposed by [8] and define a general attention model. Section 3 describes the classification of attention models. Section 4 summarizes network architectures in conjunction with the attention mechanism. Section 5 elaborates on the uses of attention in various computer vision (CV) and natural language processing (NLP) tasks. In Section 6, we discuss the interpretability that attention brings to neural networks. In Section 7, we dissect open challenges, current trends and innovative ways in the attention mechanism. In Section 8, we conclude this paper.

2. Attention mechanism

In this section, we first describe a well-known machine translation architecture using attention introduced by [8] and take it as an example to define a general attention model.

2.1. An example of attention model: RNNsearch

Bahdanau et al. [8] proposed an attention model, the *RNNsearch*, which first applied the attention mechanism to the machine translation task. The *RNNsearch* consists of a bidirectional recurrent neural network (BiRNN) [58] as an encoder and a decoder that emulates searching through a source sentence when decoding a translation, as illustrated in Fig. 3.

The encoder calculates *annotations* $(\boldsymbol{h}_1, \ldots, \boldsymbol{h}_T)$ that are the hidden state of the BiRNN based on input sequence $(x_1, \ldots, x_T)$:

$$(\boldsymbol{h_1}, \ldots, \boldsymbol{h_T}) = \text{BiRNN}(x_1, \ldots, x_T). \tag{1}$$

One shortcoming of regular RNNs is that they only utilize previous context. The BiRNN can be trained using all available input information in the past and future of a specific time frame. Specifically, as shown in Fig. 3, hidden states $\{\overrightarrow{\boldsymbol{h_1}}, \ldots, \overrightarrow{\boldsymbol{h_T}}\}$ and $\{\overleftarrow{\boldsymbol{h_1}}, \ldots, \overleftarrow{\boldsymbol{h_T}}\}$ are extracted by forward RNN and backward RNN respectively. Then the encoder obtains an annotation for one word x_i by concatenating the forward hidden state $\overrightarrow{\boldsymbol{h_i}}$ and the backward one $\overleftarrow{\boldsymbol{h_i}}$, i.e., $\boldsymbol{h_i} = [\overrightarrow{\boldsymbol{h_i}}^\top; \overleftarrow{\boldsymbol{h_i}}]^\top$.

The decoder consists of an attention block and a recurrent neural network (RNN). The function of the attention block is to compute the context vector $\boldsymbol{c}$ that represents the context relationship between the current output symbol and each term of the entire input sequence. At each time step t, the context vector $\boldsymbol{c}_t$ is computed as a weighted sum of these annotations $\boldsymbol{h}_j$:

$$\boldsymbol{c}_t = \sum_{j=1}^{T} \alpha_{tj} \boldsymbol{h}_j. \tag{2}$$

The attention weight α_{tj} of each annotation $\boldsymbol{h}_j$ is computed by

$$e_{tj} = a(\boldsymbol{s}_{t-1}, \boldsymbol{h}_j), \tag{3}$$

and

$$\alpha_{tj} = \frac{\exp(e_{tj})}{\sum_{k=1}^{T} \exp(e_{tk})}, \tag{4}$$

where a is a learnable function and it reflects the importance of the annotation $\boldsymbol{h}_j$ to the next hidden state $\boldsymbol{s}_t$ according to the state $\boldsymbol{s}_{t-1}$.

After that, the RNN outputs the most probable symbol y_t at the current step:

$$p(y_t | y_1, \ldots, y_{t-1}, \boldsymbol{x}) = \text{RNN}(\boldsymbol{c}_t). \tag{5}$$

In this way, the information of the source sentence can be distributed in the entire sequence instead of encoding all the information into a fixed-length vector through the encoder, while the

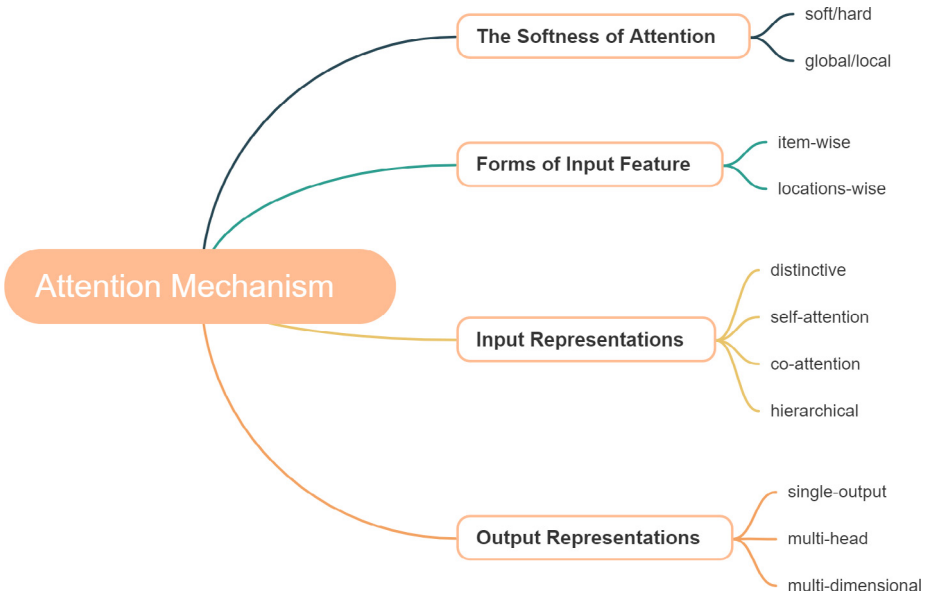


Fig. 1. Several typical approaches to attention mechanisms.

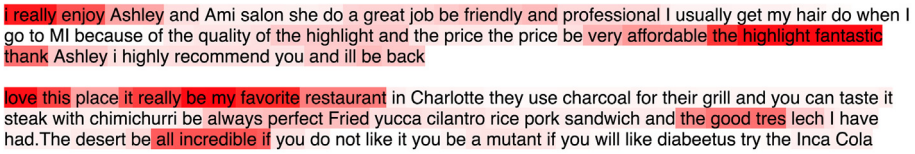


Fig. 2. Heatmap of 5 stars Yelp reviews from [30]. Heavier colors indicate higher attention weight.

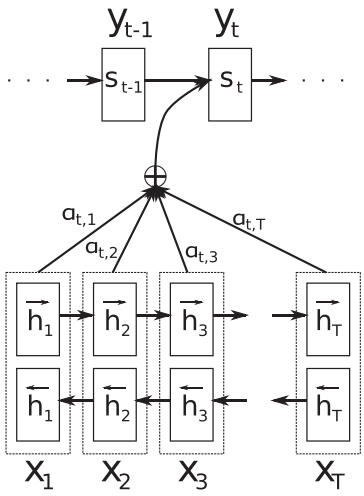


Fig. 3. Illustration of a single step of decoding in attention-based neural machine translation [8].

decoder can selectively retrieve it at each time step. This formulation enables the neural network to focus on relevant elements of the input other than on irrelevant parts.

2.2. A unified attention model

After [8] applied the attention mechanism to machine translation tasks, the attention model variants used in various application domains have evolved rapidly. Generally, the implementation process of the attention mechanism can be divided into two steps: one is to compute the attention distribution on the input information, and the other is to compute the context vector according to the attention distribution. Fig. 4 shows the unified attention model we defined, which comprises the core part shared by most of the attention models found in the surveyed literature.

When computing the attention distribution, the neural network first encodes the source data feature as $\boldsymbol{K}$, called a *key*. $\boldsymbol{K}$ can be expressed in various representations according to specific tasks and neural architectures. For instance, $\boldsymbol{K}$ may be features of a certain area of an image, word embeddings of a document, or the hidden states of RNNs, as it happens with the annotations in RNNsearch. In addition, it is usually necessary to introduce a task-related representation vector $\boldsymbol{q}$, the *query*, just like the previous hidden state of the output $\boldsymbol{s}_{t-1}$ in RNNsearch. In some cases, $\boldsymbol{q}$ can also be in the form of a matrix [16] or two vectors [59] according to specific tasks.

Then the neural network computes the correlation between queries and keys through the *score function* f (also called energy function [61] or compatibility function [62]) to obtain the *energy score* $\boldsymbol{e}$ that reflects the importance of queries with respect to keys in deciding the next output:

$$\boldsymbol{e} = f(\boldsymbol{q}, \boldsymbol{K}). \tag{6}$$

The score function f is a crucial part of the attention model because it defines how keys and queries are matched or combined. In Table 1, we list some common score functions. The two most commonly used attention mechanisms are *additive* attention (like the *alignment model* in RNNsearch) [8] and the computationally less expensive *multiplicative* (*dot-product*) attention [14]. Britz et al. [15] made an empirical comparison between these two score functions. In their experiments on the WMT'15 English→German task, they found that parameterized additive attention mechanisms slightly but consistently outperformed the multiplicative one. Moreover, Vaswani et al. [16] proposed a variant of multiplicative attention by adding the scaling factor of $\frac{1}{\sqrt{d_k}}$, where d_k is the dimension of keys. While for small values of d_k the two mechanisms perform similarly, additive attention outperforms multiplicative attention without scaling for larger values of d_k. Also, Luong et al. [14] presented general attention, concat attention, and location-based attention. *General* attention extends the concept of multiplicative attention by introducing a learnable matrix parameter $\boldsymbol{W}$, which can be applied to keys and queries with different representations. *Concat* attention aims to derive the joint representation of the keys and queries instead of comparing them. It is similar to additive attention except for computing $\boldsymbol{q}$ and $\boldsymbol{K}$ separately. In *location-based* attention, the alignment scores are solely computed from the target hidden state. In other words, the energy scores are only related to $\boldsymbol{q}$ other than $\boldsymbol{K}$. Conversely, *self*-attention

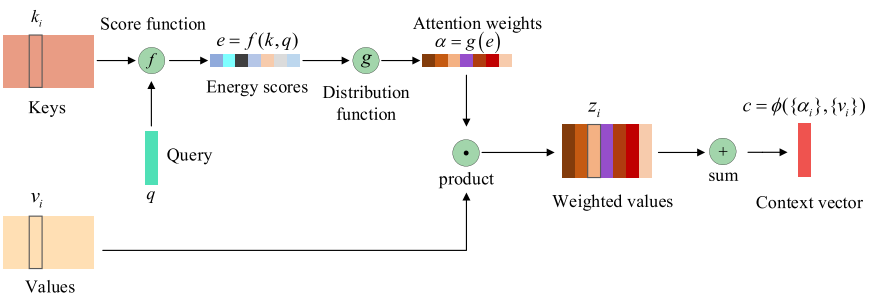


Fig. 4. The architecture of the unified attention model.

Table 1
Summary of score function f. Here, $\boldsymbol{k}$ is an element of $\boldsymbol{K}, \boldsymbol{v}, \boldsymbol{b}, \boldsymbol{W}, \boldsymbol{W_1}, \boldsymbol{W_2}$ are learnable parameters, d_k is the dimension of the input vector. The *act* is a nonlinear activation function, such as tanh and ReLU.

Name	Equation	Ref.
Additive	$f(q,k) = v^{\top} act(W_1 k + W_2 q + b)$	[15]
Multiplicative (dot-product)	$f(q,k) = q^{\top} k$	[15]
Scaled multiplicative	$f(q,k) = \frac{q^{\top} k}{\sqrt{d_k}}$	[16]
General	$f(q,k) = q^{\top} W k$	[14]
Concat	$f(q,k) = v^{\top} act(W[k;q] + b)$	[14]
Location-based	$f(q,k) = f(q)$	[14]
Similarity	$f(q,k) = \frac{q \cdot k}{\lvert q \rvert \cdot \lvert k \rvert}$	[60]

[63] is computed only based on $\boldsymbol{K}$, without the need of $\boldsymbol{q}$. Furthermore, Graves et al. [60] presented a model that compares the *similarity* between $\boldsymbol{K}$ and $\boldsymbol{q}$, which relied on cosine similarity.

After that, energy scores $\boldsymbol{e}$ are mapped to attention weights $\boldsymbol{\alpha}$ through attention distribution function g:

$$\boldsymbol{\alpha} = g(\boldsymbol{e}). \tag{7}$$

The distribution function g corresponds to the softmax in RNNsearch, which normalizes all the energy scores to a probability distribution. In addition to softmax, some researchers have tried other distribution functions. A limitation of the softmax function is that the resulting probability distribution always has full support, *i.e.*, $\text{softmax}(\boldsymbol{z}) > 0$ for every term of $\boldsymbol{z}$. This is a disadvantage in applications where a sparse probability distribution is desired, in which case [64] proposed *sparsemax* that may assign exactly zero probability to some of its output variables. Besides, [65] used another distribution function, the logistic sigmoid, which scaled energy scores between 0 and 1. They also compared sigmoid with softmax in their experiments, and the results showed that sigmoid function performed better or worse in different tasks.

When the neural network computes context vectors, it is often necessary to introduce a new data feature representation $\boldsymbol{V}$, called *value*. Each element of $\boldsymbol{V}$ corresponds to one and only one element of $\boldsymbol{K}$. In many architectures, the two are the same representation of input data, just like the annotations in RNNsearch. Based on previous work [66–68], Daniluk et al. [69] hypothesized that such overloaded use of these representations makes it difficult to train the model, so they proposed a modification to the attention mechanism which separates these functions explicitly. They used different representations of the input to compute the attention distribution and the contextual information. In other words, $\boldsymbol{V}$ and $\boldsymbol{K}$ are different representations of the same data in their *key-value pair* attention mechanism. In particular, $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ are three different representations of the same data in the self-attention mechanism [16].

Once attention weights and values are computed, the context vector $\boldsymbol{c}$ is computed by:

$$\boldsymbol{c} = \phi(\{\alpha_i\}, \{\boldsymbol{v}_i\}), \tag{8}$$

where ϕ is a function that returns a single vector given the set of values and their corresponding weights.

The common implementation of the function ϕ is to perform a weighted sum of V:

$$\boldsymbol{z}_i = \alpha_i \boldsymbol{v}_i, \tag{9}$$

and

$$\boldsymbol{c} = \sum_{i=1}^{n} \boldsymbol{z}_i, \tag{10}$$

where $\boldsymbol{z_i}$ is a weighted representation of an element in values and n is the dimension of $\boldsymbol{Z}$. Besides, there is another way to implement the function ϕ, which will be elaborated in Section 3.1. Either way, the context vector will be determined primarily due to the higher attention weight associated with the value.

The above is our description of the common architectures in the attention model. Here we quote from Vaswani et al. [16], the attention mechanism "can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.".

In addition, we would like to discuss the performance indicator for evaluating attention mechanisms. Generally, attention mechanisms in deep learning are attached to neural network models to enhance their ability to process information. Therefore, it is hard to evaluate the performance of attention mechanism without deep learning models. A common approach is ablation study that means to analyze the performance gap between models with/without the attention mechanism. Besides, the attention mechanism can be evaluated by visualizing the attention weight (as shown in Fig. 2), but this manner cannot be quantified.

3. Taxonomy of attention

In the previous section, we summarized a general attention model and explained each step of attention mechanism implementation in detail. As a method to improve the information processing ability of neural network, attention mechanism can be applied to most models in various fields of deep learning. Although the principle of attention models is the same, researchers have made some modifications and improvements to the attention mechanisms in order to better adapt them to specific tasks. We categorize attention mechanisms according to four criteria (as shown in Table 2). In this section, we elaborate on different types of attention mechanisms in each criterion through reviewing several seminal papers. In addition, we would like to emphasize that attention mechanisms in different criteria are not mutually exclusive, so there may be a combination of multiple criteria in an attention model (see Section 5 and Table 3).

3.1. The softness of attention

The attention proposed by Bahdanau et al. [8] as mentioned above belongs to *soft* (deterministic) attention, which uses a weighted average of all keys to building the context vector. For the soft attention, the attention module is differentiable with respect to the inputs, so the whole system can still be trained by standard back-propagation methods.

Correspondingly, the *hard* (stochastic) attention was proposed by Xu et al. [9], in which the context vector is computed from stochastically sampled keys. In this way, the function ϕ in Eq. (8) can be implemented as the following ones:

$$\tilde{\alpha} \sim Multinoulli(\{\alpha_i\}), \tag{11}$$

and

Table 2
Four criteria for categorizing attention mechanism, and types of attention within each criterion.

Criterion	Type
The Softness of Attention	Aoft/hard, global/local
Forms of Input Feature	Item-wise, location-wise
Input Representations	Distinctive, self, co-attention, hierarchical
Output Representations	Single-output, multi-head, multi-dimensional

Table 3
Examples of combinations between different categories.

Reference	Application	Category			
		The Softness of Attention	Forms of Input Feature	Input Representations	Output Representations
[8]	Machine Translation	Soft	Item-wise	Distinctive	Single-output
[7]	Image Classification	Hard	Location-wise	Distinctive	Single-output
[16]	Machine Translation	Soft	Item-wise	Distinctive	Multi-head
[75]	Visual Question Answering	Soft	Item-wise	Co-attention & Hierarchical	Single-output
[91]	Language Understanding	Soft	Item-wise	Distinctive	Multi-dimensional
[73]	Image Classification	Soft	Location-wise	Distinctive	Single-output
[63]	Document Classification	Soft	Item-wise	Hierarchical	Single-output

$$\boldsymbol{c} = \sum_{i=1}^{n} \tilde{\alpha}_i \boldsymbol{v}_i. \tag{12}$$

Compared with soft attention model, hard attention model is computationally less expensive because it does not need to calculate attention weights of all elements at each time. However, making a hard decision at each position of the input feature will make the module non-differentiable and difficult to optimize, so the whole system can be trained by maximizing an approximate variational lower bound or equivalently by REINFORCE [70].

On this basis, Luong et al. [14] presented the *global* attention and *local* attention mechanism for machine translation. The global attention is similar to soft attention. The local attention can be viewed as an interesting blend between the hard and soft attention, in which only a subset of source words are considered at a time. This approach is computationally less expensive than global attention or soft attention. At the same time, unlike hard attention, this approach is differentiable almost everywhere, making it easier to implement and train.

3.2. Forms of input feature

The attention mechanisms can be divided into *item-wise* and *location-wise* according to whether the input feature is a sequence of items. The *item-wise* attention requires that the input is either explicit items or an additional preprocessing step is added to generate a sequence of items from the source data. For instance, the item can be a single word embedding in RNNsearch [8], or a single feature map in SENet [71]. In the attention model, the encoder encodes each item as a separate code, and assigns different weights to them during decoding. On the contrary, *location-wise* attention is aimed at tasks that are difficult to obtain distinct input items, and generally such attention mechanism is used in visual tasks. For example, the decoder processes the multi-resolution crop of the input image at each step [7,72], or transforms the task-related region into a canonical, expected pose to simplify inference in the subsequent layers [73].

Another difference is the way it is calculated when combined with soft/hard attention mechanism. The item-wise soft attention calculates a weight for each item, and then makes a linear combination of them. The location-wise soft attention accepts an entire feature map as input and generates a transformed version through the attention module. Instead of a linear combination of all items, the item-wise hard attention stochastically picks one or some items based on their probabilities. The location-wise hard attention stochastically picks a sub-region as input and the location of the sub-region to be picked is calculated by the attention module.

3.3. Input representation

There are two features about input representation in most of the attention models mentioned above: 1) These models include a single input and corresponding output sequence; 2) The keys and queries belong to two independent sequences. This kind of attention is called *distinctive* attention [74]. In addition, the attention mechanism has many different forms of input representations (Fig. 5).

Lu et al. [75] presented a multi-inputs attention model for visual question answering task, the *co-attention*, that jointly reasons about image and question attentions. Co-attention can be performed parallelly or alternatively. The former generates image and question attention simultaneously, while the latter sequentially alternates between generating image and question attentions. Furthermore, co-attention can be coarse-grained or fine-grained [76]. Coarse-grained attention computes attention on each input, using an embedding of the other input as a query. Fine-grained attention evaluates how each element of an input affects each element of the other input. Co-attention has been used successfully in a variety of tasks including sentiment classification [77], text matching [78], named entity recognition [79], entity disambiguation [80], emotion cause analysis [81] and sentiment classification [82].

Wang et al. [83] presented *self* (*inner*) attention which computes attention only based on the input sequence. In other words, the query, key and value are different representations of the same input sequence. Such a model has proven to be effective by several authors, who have exploited it in different fashions [16,30,84–87]. A well-known application is Transformer [16], the first sequence transduction model based entirely on self-attention without RNNs. Applications of this model in various fields will be described in Section 5.

Attention weight can be computed not only from the original input sequence but also from different abstraction levels, which we refer to as *hierarchical* attention. Yang et al. [63] proposed a hierarchical attention network (HAM) for document classification, which has two levels of attention mechanisms: the word-level and sentence-level. The hierarchical attention allows the HAM to aggregate important words into a sentence and then aggregate important sentences to a document. Furthermore, the hierarchy can be extended further. Wu et al. [88] added a user level on top, applying attention also at the document level. Contrary to the above model weight learning from lower level to higher level, the model proposed by Zhao and Zhang [61] also used hierarchical attention but the attention weight is learned from the higher level to lower level. Except for natural language processing, hierarchical attention is also used in computer vision. For instance, Xiao et al. [89] proposed a two-level attention method, which is object-level and part-level attention. It is the first fine-grained image classification method that does not use additional part information and only relies on the model to generate attention weight.

3.4. Output representation

In this part, we discuss various types of output representations in attention models. Among them, the common one is the *single-output* attention which refers to a single feature representation in each time step. Specifically, the energy scores are represented by one and only

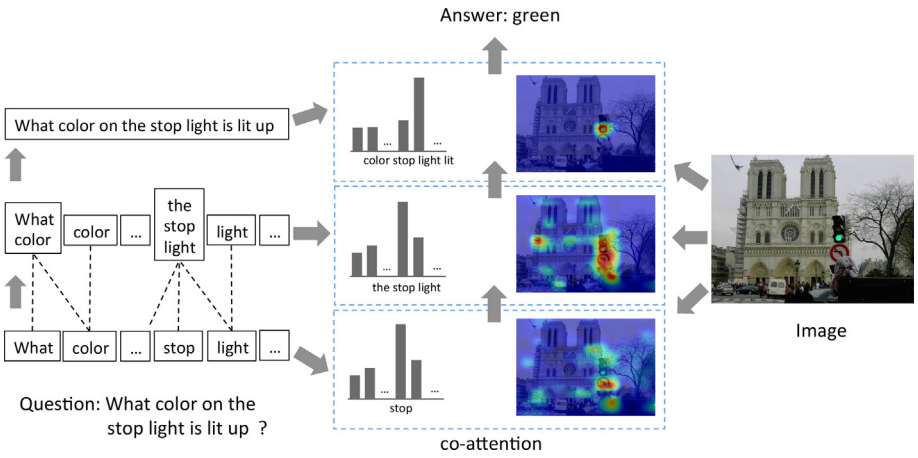
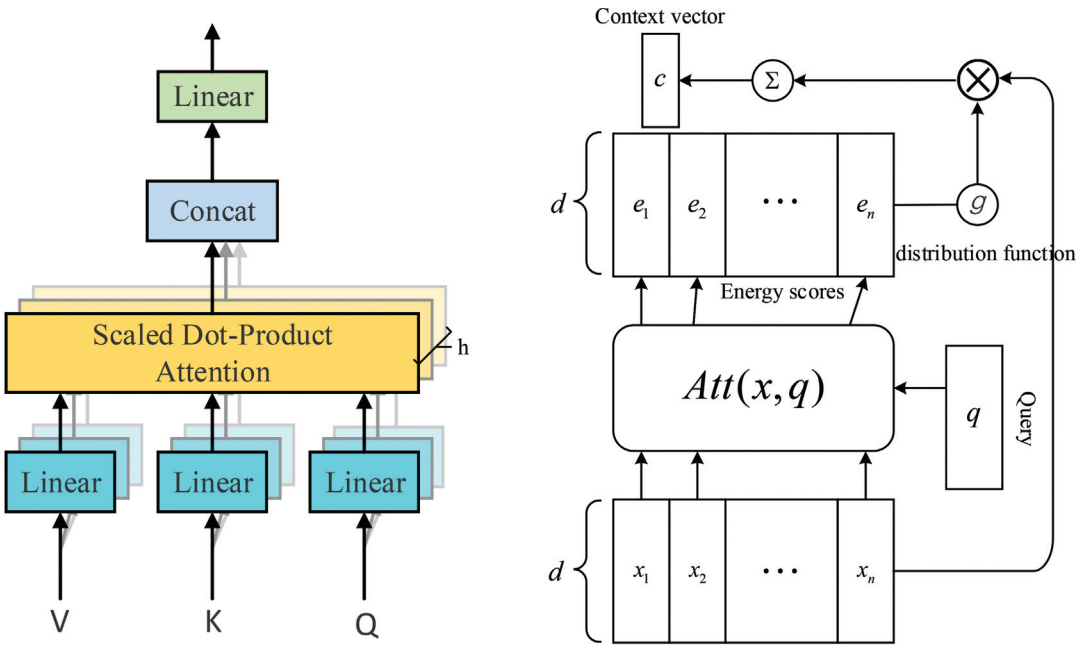


Fig. 5. An example of hierarchical attention and co-attention. Figure from [75].

one vector at each time step. However, in some cases, using a single feature representation may not be able to complete downstream tasks well. Next, we describe two multi-output attention models: multi-head and multi-dimensional as illustrated in Fig. 6.

In many applications of the convolutional neural network, it has been proved that multiple channels can express input data more comprehensively than single channel. Also in attention models, in some cases, using single attention distribution of the input sequence may not suffice for downstream tasks. Vaswani et al. [16] proposed *multi-head* attention that linearly projects the input sequence (Q, K, V) to multiple subspaces based on learnable parameters, then applies scaled dot-product attention to its representation in each subspace, and finally concatenates their output. In this way, it allows the model to jointly attend to information from different representation subspaces at different positions. Li et al. [90] proposed three disagreement regularizations to augment the multi-head attention model on the subspace, the attended positions and the output representation respectively. This approach encourages diversity among attention heads so that different heads can learn distinct features and the effectiveness is validated through translation tasks.

Another approach is the *multi-dimensional* proposed by Shen et al. [91], which computes a feature-wise score vector for keys by replacing weight scores vector with a matrix. In this way, the neural network can calculate multiple attention distributions for the same data. This is especially useful for natural language processing where word embeddings suffer from the polysemy problem. Furthermore, Lin et al. [30] and Du et al. [92] added Frobenius penalties to enforce the distinction between models of relevance, and successfully applied it to various tasks, including



(a) Multi-head attention [16] (b) Multi-dimensional attention [92]

Fig. 6. Illustrations of multi-output representations.

author profiling, sentiment classification, textual entailment, and distantly supervised relation extraction.

4. Network architectures with attention

In this section, we describe three neural network architectures used in conjunction with attention. We first elaborate on the encoder-decoder framework used by most attention models. Then we describe a special network architecture of attention models, the memory network, which is equipped with long-term external memory. Finally, we introduce special neural network structures combined with the attention mechanism, in which RNNs are not used in the process of capturing long-distance dependencies.

4.1. Encoder-decoder

The *Encoder-Decoder* (as shown in Fig. 7) is a general framework based on neural networks, which aim to handle the mapping between highly structured input and output. Here, we first briefly describe the RNN encoder-decoder framework proposed by Cho et al. [5] and Sutskever et al. [13] for machine translation tasks. This architecture consists of two RNNs that act as an encoder and a decoder pair.

The encoder maps a variable-length source sequence $\boldsymbol{x} = (x_1, \cdots, x_{T_x})$ to a fixed-length vector $\boldsymbol{c}$. The most common approach is to use an RNN such that

$$\boldsymbol{h}_t = f(x_t, \boldsymbol{h}_{t-1}), \tag{13}$$

and

$$\boldsymbol{c} = q(\{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_{T_x}\}), \tag{14}$$

where $\boldsymbol{h}_t$ is a hidden state at time step t, and $\boldsymbol{c}$ is a context vector generated from the sequence of the hidden states. f and q are non-linear activation functions. For example, f may be as simple as an element-wise logistic sigmoid function and as complex as a GRU [5] or LSTM [4], q may be as the hidden state $\boldsymbol{h}_{T_x}$ of the last time step.

The decoder is trained to generate a variable-length target sequence $\boldsymbol{y} = (y_1, \cdots, y_{T_y})$ by predicting the next symbol y_t given the hidden state $\boldsymbol{h}_t$. However, unlike the encoder, both y_t and $\boldsymbol{h}_t$ are also conditioned on y_{t-1} and the context vector $\boldsymbol{c}$. Hence, the hidden state of the decoder at each time step t is computed by:

$$\boldsymbol{h}_t = f(\boldsymbol{h}_{t-1}, y_{t-1}, \boldsymbol{c}). \tag{15}$$

Each conditional probability is modeled as

$$p(y_t|\{y_1, \cdots, y_{t-1}\}, \boldsymbol{x}) = g(y_{t-1}, \boldsymbol{h}_t, \boldsymbol{c}). \tag{16}$$

The encoder and decoder of the proposed model are jointly trained to maximize the conditional probability of a target sequence given a source sequence. A potential issue with the framework is that a neural network needs to be able to compress all the necessary information of a source data into a fixed-length vector. This may make it difficult for the neural network to cope with long sentences, especially those that are longer than the sentences in the training corpus. This problem can be alleviated by adding attention mechanism into the encoder-decoder framework:

$$\boldsymbol{c}_t = att(\{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_{T_x}\}), \tag{17}$$

and

$$p(y_t|\{y_1, \cdots, y_{t-1}\}, \boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{y_{t-1}}, \boldsymbol{h_t}, \boldsymbol{c_t}). \tag{18}$$

where *att* is the attention method mentioned in Section 2 and $\boldsymbol{c}_t$ is the context vector generated by the attention mechanism when decoding at time t. The introduction of the attention mechanism can ensure that the contributions of elements in the source sequence are different when decoding different target elements, as shown in Fig. 8.

Since this encoder-decoder framework does not limit the length of the input and output sequences, it has a wide range of applications, such as image and video captioning [9,93,94], generative dialog system [95], visual question answering [75] and speech recognition [24,96]. Moreover, encoder and decoder can also be constructed using other architectures, not necessarily only RNNs. Although the attention model can be regarded as a general idea and does not depend on a specific framework itself [7,97,73], most attention models are currently accompanied by an encoder-decoder framework.

4.2. Memory networks

In addition to the encoder-decoder framework in the previous section, the attention mechanism is also used in conjunction with memory networks. Inspired by the human brain's mechanism of handling information overload, the memory network introduces extra external memory into the neural network. Concretely, memory networks [99,60,98,100,101] save some task-related information in the auxiliary memory by introducing external auxiliary memory units and then reads it when needed, which not only effectively increases the network capacity but also improves the network computing efficiency. Compared with general attention mechanism, the memory network replaces the key with the long-term auxiliary memory and then matches the content through the attention mechanism.

A well-known example is a differentiable end-to-end memory network proposed by [98], which can read information from exter-

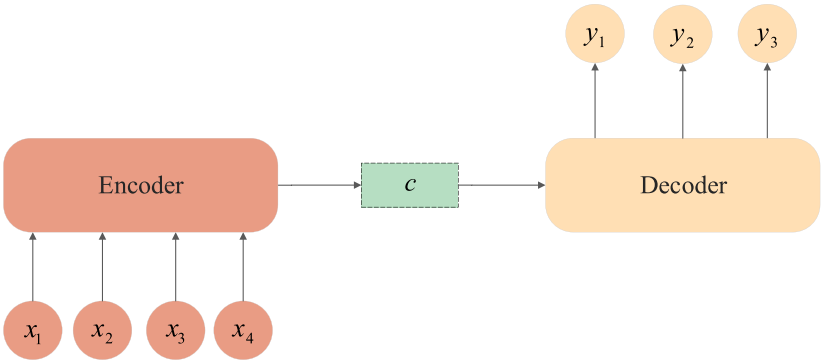


Fig. 7. Illustration of the encoder-decoder framework without the attention mechanism.

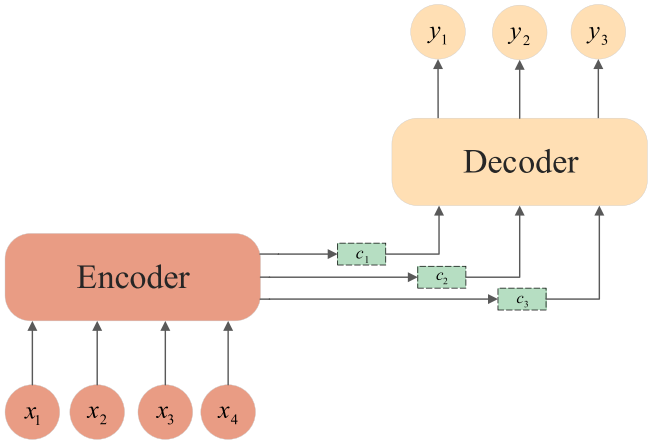


Fig. 8. Illustration of the encoder-decoder framework using the attention mechanism.

nal information multiple times. The core idea is to convert the input set into two external memory units, one for addressing, and another for output, as shown in Fig. 9,10. End-to-end memory networks can be regarded as a form of attention: the key-value pair attention mechanism. Different from the usual attention, instead of modeling attention only over a single sequence they use two external memory units to model it over a large database of sequences. In other words, we can regard the attention mechanism as an interface that separates the storage of information from the calculation, so that the network capacity can be greatly increased with a small increase in network parameters.

4.3. Networks without RNNs

As mentioned above, both the encoder and decoder in the encoder-decoder framework can be implemented in multiple ways as shown in Fig. 10. Encoder-decoder architectures based RNNs typically factor computation along with the symbol positions of the input and output sequences. This inherently sequential nature results in computational inefficiency, as the processing cannot be parallelized. On the other hand, capturing long-distance dependencies is essential because the attention mechanism in encoder-decoder framework needs to obtain contextual information. However, the computational complexity of establishing long-distance dependence for a sequence of length n through RNN is $O(n)$. In this part, we describe other implementations of the encoder-decoder framework combined with the attention mechanism, which discards RNNs.

Gehring et al. [102] proposed an encoder-decoder architecture that relied entirely on convolutional neural networks combined with the attention mechanism. In contrast to the fact that recurrent networks maintain a hidden state of the entire past, convolutional networks do not rely on the computations of the previous time step, so that it allows parallelization on each element in a sequence. This architecture enables the network to capture long-distance dependencies by stacking multiple layers of CNN, the computational complexity becomes $O(n/k)$ for a multi-layer CNN with a convolution kernel size of k. Moreover, this convolution method can discover the compositional structure in the sequence more easily because of its hierarchical representations.

Vaswani et al. [16] proposed another network architecture, the *Transformer*, which relies entirely on the self-attention mechanism to compute representations of its input and output without resorting to RNNs or CNNs. The Transformer is composed of two components: position-wise feed-forward network (FFN) layer and multi-head attention layer. Position-wise FFN is a fully connected feed-forward network, which is applied to each position separately and identically. This method can ensure the position information of each symbol in the input sequence during the operation. Multi-head attention allows the model to focus on information from different representation subspaces from different positions by stacking multiple self-attention layers, just like multiple channels of CNN. In addition to being more parallelizable, the complexity of establishing long-distance dependence through the self-attention mechanism is $O(1)$.

5. Applications

In previous sections, we have shown that attention models have been widely applied to various tasks. Here we introduce some specific applications of attention models in CV and NLP. We do not intend to provide a comprehensive review of all the neural architectures that use the attention mechanism, but focus on several general attention methods in attention models.

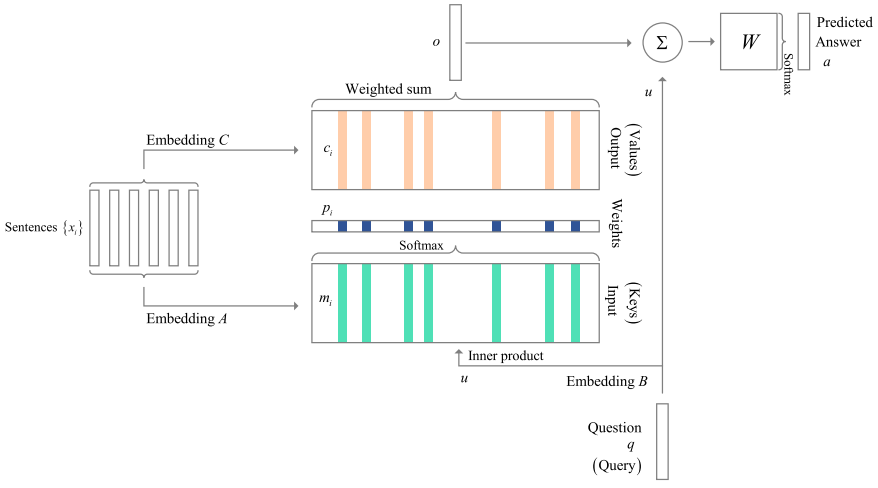
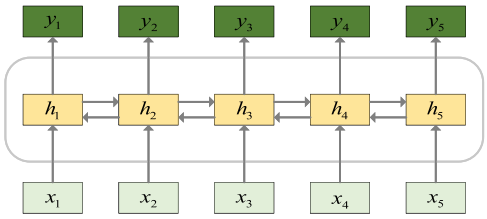
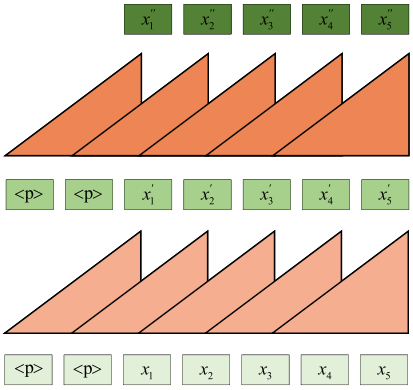


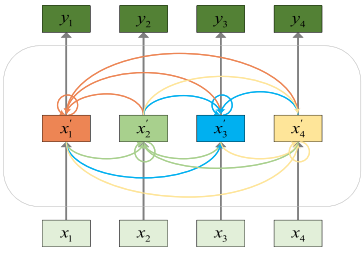
Fig. 9. A single layer version of the end-to-end memory networks [98]. Here, the question, input and output correspond to query, keys and values in the unified attention model respectively.



(a) Single-layer RNN model structure.



(b) Multi-layer CNN model structure.



(c) Single-layer self-attention model structure.

Fig. 10. Illustration of three structures used to capture long distance dependencies.

5.1. Applications in computer vision

In this part, we describe attention mechanisms in CV by introducing several typical papers on different aspects of neural architectures.

Spatial attention allows neural networks to learn the positions that should be focused on, as shown in Fig. 11. Through this attention mechanism, the spatial information in the original picture is transformed into another space and the key information is retained. Mnih et al. [7] presented a spatial attention model that is formulated as a single RNN that takes a glimpse window as its input and uses the internal state of the network to select the next location to focus on as well as to generate control signals in a dynamic environment. Jaderberg et al. [73] introduced a differentiable spatial transformer network (STN) that can find out the areas that need to be paid attention to in the feature map through transformations such as cropping, translation, rotation, scale, and skew. Unlike pooling layers, the spatial transformer module is a dynamic mechanism that can actively spatially transform an image (or a feature map) by producing an appropriate transformation for each input sample.

Channel attention allows neural networks to learn what should be focused on, as shown in Fig. 12. Hu et al. [71] proposed the squeeze-and-excitation (SE) network that adaptively recalibrated channel-wise feature responses by explicitly modeling interdependencies between channels. In that squeeze-and-excitation module, it used global average-pooled features to compute channel-wise attention. Li et al. [103] proposed the SKNet that improved the efficiency and effectiveness of object recognition by adaptive kernel selection in a channel attention manner. Besides, Stollenga et al. [104] proposed a channel hard attention mechanism that

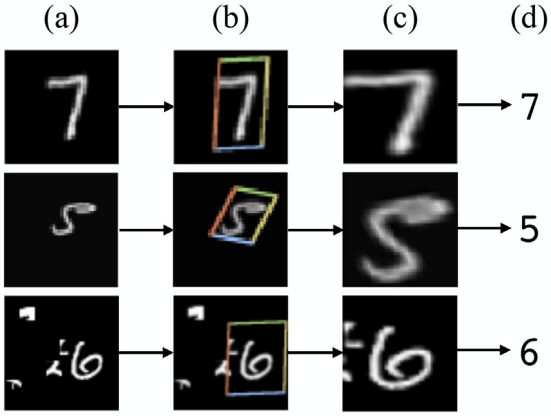


Fig. 11. An example of the spatial attention from [73].

improved classification performance by allowing the network to iteratively focus on the attention of its filters.

Capturing long-range dependencies is of central importance in deep neural networks, which is beneficial to visual understanding problems. [87] applied the self-attention mechanism to the computer vision task to solve this problem, called *non-local attention*, as shown in Fig. 13. They proposed the non-local module that got attention masks by calculating the correlation matrix between each spatial point in the feature map, then the attention guided dense contextual information to aggregate. However, this method also has the following problems: 1) Only the positional attention module is involved, not the commonly used channel attention

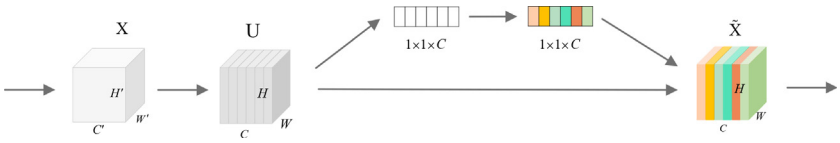


Fig. 12. Illustration of the channel attention [71].

mechanism. 2) When the input feature map is very large, there is a problem of low efficiency. Although there are other methods to solve this problem, such as scaling, it will lose information and is not the best way to deal with this problem. To address these problems, the researchers improved the non-local method and combined the channel attention to propose the *mixed attention* [105–111].

Different from previous studies, CCNet [112] used the *positional-wise attention*, which generated a huge attention map to record the relationship between each pixel-pair in the feature map, as shown in Fig. 14. The criss-cross attention module collected contextual information in horizontal and vertical directions to enhance pixel-wise representative capability. Moreover, the recurrent criss-cross attention allowed to capturing dense long-range contextual information from all pixels with less computing cost and less memory cost.

5.2. Applications in natural language processing

In this part, we first introduce some attention methods used in different tasks of NLP and then describe some common pre-training word representations implemented with the attention mechanism for NLP tasks.

Neural machine translation uses neural networks to translate text from one language to another. In the process of translation, the alignment of sentences in different languages is a crucial problem, especially for longer sentences. Bahdanau et al. [8] introduced the attention mechanism into the neural network to improve neural machine translation by selectively focusing on parts of the source sentence during translation. Afterwards, several works have been proposed as an improvement, such as local attention [14], supervised attention [113,114], hierarchical attention [61] and self attention [115,16]. They used different attention architectures to improve the alignment of sentences and enhance the performance of translation.

Text Classification aims at assigning labels to text and has broad applications including topic labeling [116], sentiment classification [117,118], and spam detection [119]. In these classification tasks, self attention is mainly used to construct more effective document representation [55,91,30]. On this basis, some works combined the self-attention mechanism with other attention methods, such as hierarchical self-attention [63] and multi-dimensional self-attention [30]. Besides, these tasks also applied attention model architectures (see Section 4) such as Transformer [120,121] and memory networks [122,123].

Text Matching is also a core research problem in NLP and information retrieval, which includes question answering, document search, entailment classification, paraphrase identification, and recommendation with reviews. Many researchers have come up with new approaches in conjunction with attention comprehension, such as memory networks [98], attention-over-attention [124], inner attention [83], structured attention [65] and co-attention [78,75].

Pre-trained word representations are a key component in many neural language understanding models. However, previous studies [125,126] only determined one embedding for the same word, which could not achieve contextual word embedding. Peters et al. [127] introduced a general approach of context-dependent representation with Bi-LSTM to solve this problem. Inspired by the Transformer model [62], the researchers proposed the bidirectional encoder representations from transformers (BERT) [128] and generative pre-training (GPT) [129–131] methods according to the encoder and decoder parts. BERT is a bidirectional language model and has the following two pre-training tasks: 1) Masked language model (MLM). It simply masks some percentage of the input tokens at random, and then predicts those masked tokens. 2) Next sentence prediction. It uses a linear binary classifier to determine whether two sentences are connected. GPT is a one-way model and its training methods roughly use the previous word to predict the next word. Experiments show large improvements when applying them to a broad range of NLP tasks.

6. Attention for interpretability

In recent years, artificial intelligence has been developed rapidly [132–137], especially in the field of deep learning. How-

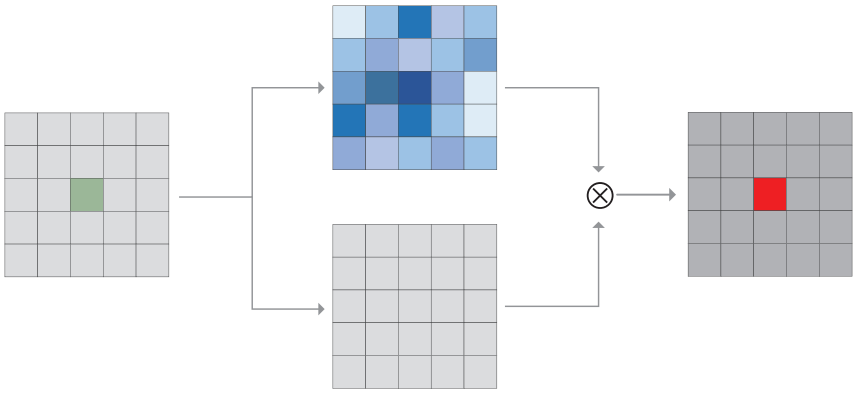


Fig. 13. Illustration of the non-local attention [87].

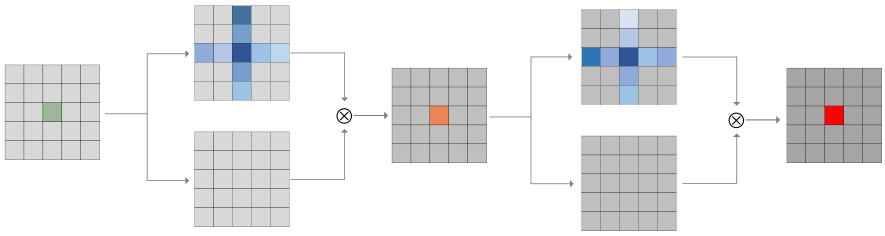
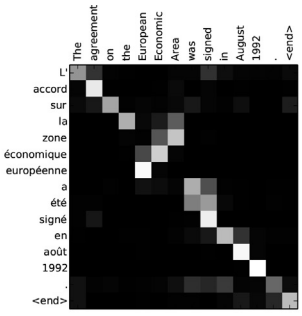
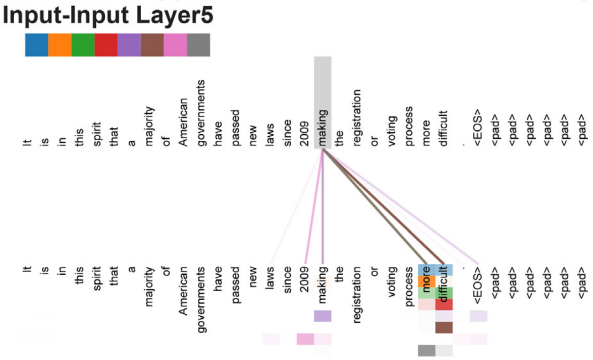


Fig. 14. Illustration of the criss-cross attention [112].



(a) Alignment of French and English sentences in machine translation [8]

(b) Relevant image regions for image captioning [9]



(c) An example of visualizing the weight of the word "making" in the Transformer model [16]

Fig. 15. Examples of visualizing the weight of attention mechanism.

ever, interpretability is a major concern for many current deep learning models. Deep learning models become increasingly complex, so it is critical to learn decision-making functions from data and ensure that we understand why a particular decision occurs [51]. The attention layer in a neural network model provides a way to reason about the model behind its predictions [54–57,138], but this is often criticized for being opaque [52,53].

As shown in Fig. 15(a), Bahdanau et al. [8] visualized attention weights annotations, which clearly show the (soft-) alignment between the words in a generated translation (French) and those in a source sentence (English). Fig. 15(b) shows the attended regions corresponding to the underlined word in the neural image caption generation process [9]. As we can see, the model learns alignments that correspond very strongly with human intuition. Furthermore, Fig. 15(c) shows an example of the visualization in the encoder self-attention layer in Transformer model proposed by Vaswani et al. [16]. Different colors represent different heads. Moreover, Voita et al. [139] evaluated the contribution made by individual attention heads to the Transformer model on translation. They discovered that different heads showed different levels of importance through pruning operations. Besides, Chan et al. [24] observed that the content-based attention mechanism could identify the start position in the audio sequence for the first character correctly.

However, some recent studies [52,53] suggested that attention could not be considered a reliable means to interpret deep neural networks. Jain and Wallace [52] performed extensive experiments across a variety of NLP tasks and proved that attention was not consistent with other explainability metrics. They find that it is very often to construct adversarial attention distributions which means that different attention weight distributions yield the equivalent predictions. Serrano and Smith [53] applied a different analysis based on intermediate representation and found that attention weights were only noisy predictors of intermediate components' importance. On the contrary, Wiegreffe and Pinter [57] made a rebuttal to their work with four alternative tests. Thus, whether or not attention is used as a means to explain neural networks is still an open topic.

7. Challenges and prospects

Attention models have become ubiquitous in deep neural networks. Since Bahdanau et al. [8] used the attention mechanism

for alignment in machine translation tasks, various attention mechanism variants have emerged endlessly. The Transformer model that only uses the self-attention mechanism proposed by Vaswani et al. [16] is also an important milestone in the attention mechanism and its variants [128,129,140–142] have been successful applied in various fields. When investigating the attention model, we find that the attention mechanism is innovative in several aspects, such as multiple implementations in the score function and distribution function, the combination of value and attention weight, and the network architecture.

There is still much room for improvement in the design of attention mechanisms. Here we summarize a few promising directions as follows.

- At present, most self-attention models represent query and key independently, however, some recent studies [143,144] have achieved good performance by combining query and key. Whether query and key are necessary to exist independently in self-attention is still an open question.
- Attention distribution function has a great influence on the computational complexity of the whole attention model. Some recent studies [145,146] show that the complexity of attention computation may be further reduced by improving the attention distribution function.
- How attention techniques developed in one area can be applied to other areas is also an interesting direction. For example, when the attention method with self-attention in NLP is applied in CV, it improves performance while also reducing efficiency, like non-local network [87].
- The combination of adaptive mechanism and attention mechanism may automatically achieve the effect of hierarchical attention without manually designing the structure of each layer.
- To explore more effective performance indicators for evaluating attention mechanism is also an interesting topic. Sen et al. [147] designed a set of evaluation methods to quantify similarities between human and attention-based neural models using novel attention-map similarity metrics. This is also an open direction for future research.

8. Conclusion

In this paper, we illustrate how the unified attention model works, and describe in detail the classification of attention mechanisms. Then we summarize network architectures used in conjunction with the attention mechanism, and introduced typical applications of attention mechanisms used in computer vision and natural language processing. Finally, we discuss the interpretability that attention mechanisms provide for the model generation process, the challenges and prospects of current attention models. In conclusion, attention mechanism has been successfully used in various fields of deep learning applications, but there are still many interesting questions to be explored.

CRediT authorship contribution statement

Zhaoyang Niu: Investigation, Methodology, Formal analysis, Writing - review & editing. **Guoqiang Zhong:** Conceptualization, Methodology, Investigation, Formal analysis, Writing - review & editing, Project administration, Supervision. **Hui Yu:** Conceptualization, Investigation, Methodology, Writing - review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the National Key Research and Development Program of China under Grant No. 2018AAA0100400, the Joint Fund of the Equipments Pre-Research and Ministry of Education of China under Grant No. 6141A020337, the Natural Science Foundation of Shandong Province under Grant No. ZR2020MF131, and the Science and Technology Program of Qingdao under Grant No. 21-1-4-ny-19-nsh.

References

[1] R.A. Rensink, The dynamic representation of scenes, Visual Cogn. 7 (2000) 17–42.
[2] M. Corbetta, G.L. Shulman, Control of goal-directed and stimulus-driven attention in the brain, Nat. Rev. Neurosci. 3 (2002) 201–215.
[3] J.K. Tsotsos, S.M. Culhane, W.Y.K. Wai, Y. Lai, N. Davis, F. Nuflo, Modeling visual attention via selective tuning, Artif. Intell. 78 (1995) 507–545.
[4] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780.
[5] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, in: EMNLP, ACL, 2014, pp. 1724–1734..
[6] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 1254–1259.
[7] V. Mnih, N. Heess, A. Graves, K. Kavukcuoglu, Recurrent models of visual attention, in: NIPS, pp. 2204–2212..
[8] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: ICLR..
[9] K. Xu, J. Ba, R. Kiros, K. Cho, A.C. Courville, R. Salakhutdinov, R.S. Zemel, Y. Bengio, Show, attend and tell: neural image caption generation with visual attention, in: ICML, Volume 37 of JMLR Workshop and Conference Proceedings, JMLR.org, 2015, pp. 2048–2057..
[10] J. Lu, C. Xiong, D. Parikh, R. Socher, Knowing when to look: Adaptive attention via a visual sentinel for image captioning, in: CVPR, IEEE Computer Society, 2017, pp. 3242–3250.
[11] G. Liu, J. Guo, Bidirectional LSTM with attention mechanism and convolutional layer for text classification, Neurocomputing 337 (2019) 325–338.
[12] Y. Li, L. Yang, B. Xu, J. Wang, H. Lin, Improving user attribute classification with text and social network attention, Cogn. Comput. 11 (2019) 459–468.
[13] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: NIPS, pp. 3104–3112..
[14] T. Luong, H. Pham, C.D. Manning, Effective approaches to attention-based neural machine translation, in: EMNLP, The Association for Computational Linguistics, 2015, pp. 1412–1421..
[15] D. Britz, A. Goldie, M. Luong, Q.V. Le, Massive exploration of neural machine translation architectures, CoRR abs/1703.03906 (2017).
[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: NIPS, pp. 5998–6008..
[17] S. Song, C. Lan, J. Xing, W. Zeng, J. Liu, An end-to-end spatio-temporal attention model for human action recognition from skeleton data, in: AAAI, AAAI Press, 2017, pp. 4263–4270..
[18] Y. Tian, W. Hu, H. Jiang, J. Wu, Densely connected attentional pyramid residual network for human pose estimation, Neurocomputing 347 (2019) 13–23.
[19] A. Zhao, L. Qi, J. Li, J. Dong, H. Yu, LSTM for diagnosis of neurodegenerative diseases using gait data, in: H. Yu, J. Dong (Eds.), Ninth International Conference on Graphic and Image Processing (ICGIP 2017), vol. 10615, p. 106155B..
[20] P. Zhang, J. Xue, C. Lan, W. Zeng, Z. Gao, N. Zheng, Adding attentiveness to the neurons in recurrent neural networks, in: ECCV (9), Volume 11213 of Lecture Notes in Computer Science, Springer, 2018, pp. 136–152..
[21] K. Song, T. Yao, Q. Ling, T. Mei, Boosting image sentiment analysis with visual attention, Neurocomputing 312 (2018) 218–228.
[22] X. Yan, S. Hu, Y. Mao, Y. Ye, H. Yu, Deep multi-view learning methods: a review, Neurocomputing (2021).
[23] J. Chorowski, D. Bahdanau, K. Cho, Y. Bengio, End-to-end continuous speech recognition using attention-based recurrent NN: first results, CoRR abs/1412.1602 (2014)..

[24] W. Chan, N. Jaitly, Q. V. Le, O. Vinyals, Listen, attend and spell: a neural network for large vocabulary conversational speech recognition, in: ICASSP, IEEE, 2016, pp. 4960–4964.
[25] M. Sperber, J. Niehues, G. Neubig, S. Stüker, A. Waibel, Self-attentional acoustic models, in: INTERSPEECH, ISCA, 2018, pp. 3723–3727..
[26] S. Wang, L. Hu, L. Cao, X. Huang, D. Lian, W. Liu, Attention-based transactional context embedding for next-item recommendation, in: AAAI, AAAI Press, 2018, pp. 2532–2539.
[27] H. Ying, F. Zhuang, F. Zhang, Y. Liu, G. Xu, X. Xie, H. Xiong, J. Wu, Sequential recommender system based on hierarchical attention networks, in: IJCAI, ijcai.org, 2018, pp. 3926–3932..
[28] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: ICLR (Poster), OpenReview.net, 2018..
[29] K. Xu, L. Wu, Z. Wang, Y. Feng, V. Sheinin, Graph2seq: graph to sequence learning with attention-based neural networks, CoRR abs/1804.00823 (2018)..
[30] Z. Lin, M. Feng, C.N. dos Santos, M. Yu, B. Xiang, B. Zhou, Y. Bengio, A structured self-attentive sentence embedding, in: ICLR (Poster), OpenReview.net, 2017..
[31] K. Zhang, G. Zhong, J. Dong, S. Wang, Y. Wang, Stock market prediction based on generative adversarial network, in: R. Bie, Y. Sun, J. Yu (Eds.), 2018 International Conference on Identification, Information and Knowledge in the Internet of Things, IIKI 2018, Beijing, China, October 19–21, 2018, Volume 147 of Procedia Computer Science, Elsevier, 2018, pp. 400–406..
[32] C. Ieracitano, A. Paviglianiti, M. Campolo, A. Hussain, E. Pasero, F.C. Morabito, A novel automatic classification system based on hybrid unsupervised and supervised machine learning for electrospun nanofibers, IEEE CAA J. Autom. Sinica 8 (2021) 64–76.
[33] Z. Fan, G. Zhong, H. Li, A feature fusion network for multi-modal mesoscale eddy detection, in: H. Yang, K. Pasupa, A.C. Leung, J.T. Kwok, J.H. Chan, I. King (Eds.), Neural Information Processing – 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 23–27, 2020, Proceedings, Part I, volume 12532 of Lecture Notes in Computer Science, Springer, 2020, pp. 51–61..
[34] H. Yu, O. Garrod, R. Jack, P. Schyns, A framework for automatic and perceptually valid facial expression generation, Multimedia Tools Appl. 74 (2015) 9427–9447.
[35] Q. Li, Z. Fan, G. Zhong, Bednet: bi-directional edge detection network for ocean front detection, in: H. Yang, K. Pasupa, A.C. Leung, J.T. Kwok, J.H. Chan, I. King (Eds.), Neural Information Processing – 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 18–22, 2020, Proceedings, Part IV, volume 1332 of Communications in Computer and Information Science, Springer, 2020, pp. 312–319..
[36] Z. Fan, G. Zhong, H. Wei, H. Li, Ednet: a mesoscale eddy detection network with multi-modal data, in: 2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19–24, 2020, IEEE, 2020, pp. 1–7..
[37] X. Liu, Y. Xia, H. Yu, J. Dong, M. Jian, T.D. Pham, Region based parallel hierarchy convolutional neural network for automatic facial nerve paralysis evaluation, IEEE Trans. Neural Syst. Rehab. Eng. 28 (2020) 2325–2332.
[38] W. Yue, Z. Wang, W. Liu, B. Tian, S. Lauria, X. Liu, An optimally weighted user- and item-based collaborative filtering approach to predicting baseline data for friedreich's ataxia patients, Neurocomputing 419 (2021) 287–294.
[39] N. Zeng, H. Li, Z. Wang, W. Liu, S. Liu, F.E. Alsaadi, X. Liu, Deep-reinforcement-learning-based images segmentation for quantitative analysis of gold immunochromatographic strip, Neurocomputing (2020).
[40] W. Liu, Z. Wang, X. Liu, N. Zeng, D. Bell, A novel particle swarm optimization approach for patient clustering from emergency departments, IEEE Trans. Evol. Comput. 23 (2019) 632–644.
[41] N. Zeng, Z. Wang, H. Zhang, K.-E. Kim, Y. Li, X. Liu, An improved particle filter with a novel hybrid proposal distribution for quantitative analysis of gold immunochromatographic strips, IEEE Trans. Nanotechnol. 18 (2019) 819–829.
[42] Y. Ming, X. Meng, C. Fan, H. Yu, Deep learning for monocular depth estimation: a review, Neurocomputing 438 (2021) 14–33.
[43] Y. Xia, H. Yu, F. Wang, Accurate and robust eye center localization via fully convolutional networks, IEEE CAA J. Autom. Sinica 6 (2019) 1127–1138.
[44] Y. Guo, Y. Xia, J. Wang, H. Yu, R. Chen, Real-time facial affective computing on mobile devices, Sensors 20 (2020) 870.
[45] Y. Wang, X. Dong, G. Li, J. Dong, H. Yu, Cascade regression-based face frontalization for dynamic facial expression analysis, Cogn. Comput. (2021) 1–14.
[46] X. Zhang, D. Ma, H. Yu, Y. Huang, P. Howell, B. Stevens, Scene perception guided crowd anomaly detection, Neurocomputing 414 (2020) 291–302.
[47] A. Roy, B. Banerjee, A. Hussain, S. Poria, Discriminative dictionary design for action classification in still images and videos, Cogn. Comput. (2021).
[48] S. Liu, Y. Xia, Z. Shi, H. Yu, Z. Li, J. Lin, Deep learning in sheet metal bending with a novel theory-guided deep neural network, IEEE/CAA J. Autom. Sinica 8 (2021) 565–581.
[49] F. Luque Snchez, I. Hupont, S. Tabik, F. Herrera, Revisiting crowd behaviour analysis through deep learning: taxonomy, anomaly detection, crowd emotions, datasets, opportunities and prospects, Inf. Fusion 64 (2020) 318–335.
[50] X. Zhang, X. Yang, W. Zhang, G. Li, H. Yu, Crowd emotion evaluation based on fuzzy inference of arousal and valence, Neurocomputing 445 (2021) 194–205.
[51] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (2019) 93:1–93:42..
[52] S. Jain, B.C. Wallace, Attention is not explanation, in: NAACL-HLT (1), Association for Computational Linguistics, 2019, pp. 3543–3556..
[53] S. Serrano, N.A. Smith, Is attention interpretable?, in: ACL (1), Association for Computational Linguistics, 2019, pp. 2931–2951..
[54] L.H. Li, M. Yatskar, D. Yin, C. Hsieh, K. Chang, What does BERT with vision look at?, in: ACL, Association for Computational Linguistics, 2020, pp. 5265–5275..
[55] G. Letarte, F. Paradis, P. Giguère, F. Laviolette, Importance of self-attention for sentiment analysis, in: BlackboxNLP@EMNLP, Association for Computational Linguistics, 2018, pp. 267–275..
[56] S. Vashishth, S. Upadhyay, G.S. Tomar, M. Faruqui, Attention interpretability across NLP tasks, CoRR abs/1909.11218 (2019).
[57] S. Wiegreffe, Y. Pinter, Attention is not not explanation, in: EMNLP/IJCNLP (1), Association for Computational Linguistics, 2019, pp. 11–20..
[58] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Trans. Signal Process. 45 (1997) 2673–2681.
[59] A. Sordoni, P. Bachman, Y. Bengio, Iterative alternating neural attention for machine reading, CoRR abs/1606.02245 (2016)..
[60] A. Graves, G. Wayne, I. Danihelka, Neural turing machines, CoRR abs/1410.5401 (2014).
[61] S. Zhao, Z. Zhang, Attention-via-attention neural machine translation, in: AAAI, AAAI Press, 2018, pp. 563–570..
[62] A. Galassi, M. Lippi, P. Torroni, Attention, please! A critical review of neural attention models in natural language processing, CoRR abs/1902.02181 (2019).
[63] Z. Yang, D. Yang, C. Dyer, X. He, A.J. Smola, E.H. Hovy, Hierarchical attention networks for document classification, in: HLT-NAACL, The Association for Computational Linguistics, 2016, pp. 1480–1489..
[64] A.F.T. Martins, R.F. Astudillo, From softmax to sparsemax: A sparse model of attention and multi-label classification, in: ICML, Volume 48 of JMLR Workshop and Conference Proceedings, JMLR.org, 2016, pp. 1614–1623..
[65] Y. Kim, C. Denton, L. Hoang, A.M. Rush, Structured attention networks, arXiv: Computation and Language (2017)..
[66] A.H. Miller, A. Fisch, J. Dodge, A. Karimi, A. Bordes, J. Weston, Key-value memory networks for directly reading documents, in: EMNLP, The Association for Computational Linguistics, 2016, pp. 1400–1409..
[67] J. Ba, G.E. Hinton, V. Mnih, J.Z. Leibo, C. Ionescu, Using fast weights to attend to the recent past, in: NIPS, pp. 4331–4339..
[68] Ç. Gülçehre, S. Chandar, K. Cho, Y. Bengio, Dynamic neural turing machine with soft and hard addressing schemes, CoRR abs/1607.00036 (2016).
[69] M. Daniluk, T. Rocktäschel, J. Welbl, S. Riedel, Frustratingly short attention spans in neural language modeling, in: ICLR (Poster), OpenReview.net, 2017..
[70] R.J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Mach. Learn. 8 (1992) 229–256.
[71] J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, Squeeze-and-excitation networks, IEEE Trans. Pattern Anal. Mach. Intell. 42 (2020) 2011–2023.
[72] J. Ba, V. Mnih, K. Kavukcuoglu, Multiple object recognition with visual attention, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings..
[73] M. Jaderberg, K. Simonyan, A. Zisserman, et al., Spatial transformer networks, in: Advances in Neural Information Processing Systems, pp. 2017–2025..
[74] S. Chaudhari, G. Polatkan, R. Ramanath, V. Mithal, An attentive survey of attention models, CoRR abs/1904.02874 (2019)..
[75] J. Lu, J. Yang, D. Batra, D. Parikh, Hierarchical question-image co-attention for visual question answering, in: NIPS, pp. 289–297..
[76] F. Fan, Y. Feng, D. Zhao, Multi-grained attention network for aspect-level sentiment classification, in: EMNLP, Association for Computational Linguistics, 2018, pp. 3433–3442..
[77] W. Wang, S. J. Pan, D. Dahlmeier, X. Xiao, Coupled multi-layer attentions for co-extraction of aspect and opinion terms, in: AAAI, AAAI Press, 2017, pp. 3316–3322.
[78] Y. Tay, A.T. Luu, S.C. Hui, Hermitian co-attention networks for text matching in asymmetrical domains, in: IJCAI, ijcai.org, 2018, pp. 4425–4431..
[79] Q. Zhang, J. Fu, X. Liu, X. Huang, Adaptive co-attention network for named entity recognition in tweets, in: AAAI, AAAI Press, 2018, pp. 5674–5681.
[80] F. Nie, Y. Cao, J. Wang, C. Lin, R. Pan, Mention and entity description co-attention for entity disambiguation, in: AAAI, AAAI Press, 2018, pp. 5908–5915.
[81] X. Li, K. Song, S. Feng, D. Wang, Y. Zhang, A co-attention neural network model for emotion cause analysis with emotional context awareness, in: EMNLP, Association for Computational Linguistics, 2018, pp. 4752–4757..
[82] Y. Tay, A.T. Luu, S.C. Hui, J. Su, Attentive gated lexicon reader with contrastive contextual co-attention for sentiment classification, in: EMNLP, Association for Computational Linguistics, 2018, pp. 3443–3453..
[83] B. Wang, K. Liu, J. Zhao, Inner attention based recurrent neural networks for answer selection, in: ACL (1), The Association for Computer Linguistics, 2016..
[84] L. Wu, F. Tian, L. Zhao, J. Lai, T. Liu, Word attention for sequence to sequence text understanding, in: AAAI, AAAI Press, 2018, pp. 5578–5585.
[85] J. Pavlopoulos, P. Malakasiotis, I. Androutsopoulos, Deeper attention to abusive user content moderation, in: EMNLP, Association for Computational Linguistics, 2017, pp. 1125–1135..

[86] Z. Li, Y. Wei, Y. Zhang, Q. Yang, Hierarchical attention transfer network for cross-domain sentiment classification, in: AAAI, AAAI Press, 2018, pp. 5852–5859.
[87] X. Wang, R. B. Girshick, A. Gupta, K. He, Non-local neural networks, in: CVPR, IEEE Computer Society, 2018, pp. 7794–7803..
[88] C. Wu, F. Wu, J. Liu, Y. Huang, Hierarchical user and item representation with three-tier attention for recommendation, in: NAACL-HLT (1), Association for Computational Linguistics, 2019, pp. 1818–1826..
[89] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, Z. Zhang, The application of two-level attention models in deep convolutional neural network for fine-grained image classification, in: CVPR, IEEE Computer Society, 2015, pp. 842–850.
[90] J. Li, Z. Tu, B. Yang, M. R. Lyu, T. Zhang, Multi-head attention with disagreement regularization, in: EMNLP, Association for Computational Linguistics, 2018, pp. 2897–2903..
[91] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, C. Zhang, Disan: directional self-attention network for rnn/cnn-free language understanding, in: AAAI, AAAI Press, 2018, pp. 5446–5455..
[92] J. Du, J. Han, A. Way, D. Wan, Multi-level structured self-attentions for distantly supervised relation extraction, in: EMNLP, Association for Computational Linguistics, 2018, pp. 2216–2225..
[93] S. Venugopalan, M. Rohrbach, J. Donahue, R. J. Mooney, T. Darrell, K. Saenko, Sequence to sequence – video to text, in: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7–13, 2015, IEEE Computer Society, 2015, pp. 4534–4542..
[94] G. Ding, M. Chen, S. Zhao, H. Chen, J. Han, Q. Liu, Neural image caption generation with weighted training and reference, Cogn. Comput. 11 (2019) 763–777.
[95] X. Zhang, Q. Yang, Transfer hierarchical attention network for generative dialog system, Int. J. Autom. Comput. 16 (2019) 720–736.
[96] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, N. Jaitly, A comparison of sequence-to-sequence models for speech recognition, in: F. Lacerda (Ed.), Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20–24, 2017, ISCA, 2017, pp. 939–943.
[97] S. Wang, J. Zhang, C. Zong, Learning sentence representation with guidance of human attention, in: IJCAI, ijcai.org, 2017, pp. 4137–4143..
[98] S. Sukhbaatar, A. Szlam, J. Weston, R. Fergus, End-to-end memory networks, in: NIPS, pp. 2440–2448..
[99] J. Weston, S. Chopra, A. Bordes, Memory networks, in: ICLR..
[100] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, R. Socher, Ask me anything: Dynamic memory networks for natural language processing, in: ICML, Volume 48 of JMLR Workshop and Conference Proceedings, JMLR.org, 2016, pp. 1378–1387..
[101] M. Henaff, J. Weston, A. Szlam, A. Bordes, Y. LeCun, Tracking the world state with recurrent entity networks, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings, OpenReview.net, 2017..
[102] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y.N. Dauphin, Convolutional sequence to sequence learning, in: ICML, Volume 70 of Proceedings of Machine Learning Research, PMLR, 2017, pp. 1243–1252..
[103] X. Li, W. Wang, X. Hu, J. Yang, Selective kernel networks, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16–20, 2019, Computer Vision Foundation/ IEEE, 2019, pp. 510–519..
[104] M.F. Stollenga, J. Masci, F.J. Gomez, J. Schmidhuber, Deep networks with internal selective attention through feedback connections, in: Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8–13 2014, Montreal, Quebec, Canada, pp. 3545–3553..
[105] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, H. Lu, Dual attention network for scene segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16–20, 2019, Computer Vision Foundation/ IEEE, 2019, pp. 3146–3154..
[106] Y. Yuan, J. Wang, Ocnet: object context network for scene parsing, CoRR abs/1809.00916 (2018)..
[107] H. Zhao, Y. Zhang, S. Liu, J. Shi, C.C. Loy, D. Lin, J. Jia, Psanet: point-wise spatial attention network for scene parsing, in: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.), Computer Vision – ECCV 2018–15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part IX, volume 11213 of Lecture Notes in Computer Science, Springer, 2018, pp. 270–286..
[108] Y. Cao, J. Xu, S. Lin, F. Wei, H. Hu, Gcnet: non-local networks meet squeeze-excitation networks and beyond, in: 2019 IEEE/CVF International Conference on Computer Vision Workshops, ICCV Workshops 2019, Seoul, Korea (South), October 27–28, 2019, IEEE, 2019, pp. 1971–1980..
[109] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, X. Tang, Residual attention network for image classification, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21–26, 2017, IEEE Computer Society, 2017, pp. 6450–6458..
[110] K. Yue, M. Sun, Y. Yuan, F. Zhou, E. Ding, F. Xu, Compact generalized non-local network, in: S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada, pp. 6511–6520..
[111] S. Woo, J. Park, J. Lee, I.S. Kweon, CBAM: convolutional block attention module, in: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.), Computer Vision – ECCV 2018–15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VII, Volume 11211 of Lecture Notes in Computer Science, Springer, 2018, pp. 3–19..
[112] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, W. Liu, Ccnet: criss-cross attention for semantic segmentation, in: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27–November 2, 2019, IEEE, 2019, pp. 603–612..
[113] H. Mi, Z. Wang, A. Ittycheriah, Supervised attentions for neural machine translation, in: J. Su, X. Carreras, K. Duh (Eds.), Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016, The Association for Computational Linguistics, 2016, pp. 2283–2288..
[114] L. Liu, M. Utiyama, A.M. Finch, E. Sumita, Neural machine translation with supervised attention, in: N. Calzolari, Y. Matsumoto, R. Prasad (Eds.), COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11–16, 2016, Osaka, Japan, ACL, 2016, pp. 3093–3102..
[115] B. Yang, Z. Tu, D.F. Wong, F. Meng, L.S. Chao, T. Zhang, Modeling localness for self-attention networks, in: E. Riloff, D. Chiang, J. Hockenmaier, J. Tsujii (Eds.), Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31–November 4, 2018, Association for Computational Linguistics, 2018, pp. 4449–4458.
[116] S.I. Wang, C.D. Manning, Baselines and bigrams: simple, good sentiment and topic classification, in: The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8–14, 2012, Jeju Island, Korea – Volume 2: Short Papers, The Association for Computer Linguistics, 2012, pp. 90–94..
[117] A.L. Maas, R.E. Daly, P.T. Pham, D. Huang, A.Y. Ng, C. Potts, Learning word vectors for sentiment analysis, in: D. Lin, Y. Matsumoto, R. Mihalcea (Eds.), The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19–24 June, 2011, Portland, Oregon, USA, The Association for Computer Linguistics, 2011, pp. 142–150..
[118] B. Pang, L. Lee, Opinion mining and sentiment analysis, Found. Trends Inf. Retr. 2 (2007) 1–135.
[119] M. Sahami, S. Dumais, D. Heckerman, E. Horvitz, A bayesian approach to filtering junk e-mail, in: Learning for Text Categorization: Papers from the 1998 workshop, vol. 62, Madison, Wisconsin, pp. 98–105..
[120] Y. Song, J. Wang, T. Jiang, Z. Liu, Y. Rao, Attentional encoder network for targeted sentiment classification, CoRR abs/1902.09314 (2019).
[121] A. Ambartsoumian, F. Popowich, Self-attention: a better building block for sentiment analysis neural network classifiers, in: A. Balahur, S.M. Mohammad, V. Hoste, R. Klinger (Eds.), Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@EMNLP 2018, Brussels, Belgium, October 31, 2018, Association for Computational Linguistics, 2018, pp. 130–139..
[122] D. Tang, B. Qin, T. Liu, Aspect level sentiment classification with deep memory network, in: J. Su, X. Carreras, K. Duh (Eds.), Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1–4, 2016, The Association for Computational Linguistics, 2016, pp. 214–224..
[123] P. Zhu, T. Qian, Enhanced aspect level sentiment classification with auxiliary memory, in: E.M. Bender, L. Derczynski, P. Isabelle (Eds.), Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20–26, 2018, Association for Computational Linguistics, 2018, pp. 1077–1087..
[124] Y. Cui, Z. Chen, S. Wei, S. Wang, T. Liu, G. Hu, Attention-over-attention neural networks for reading comprehension, in: R. Barzilay, M. Kan (Eds.), Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30–August 4, Volume 1: Long Papers, Association for Computational Linguistics, 2017, pp. 593–602..
[125] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: Y. Bengio, Y. LeCun (Eds.), 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2–4, 2013, Workshop Track Proceedings..
[126] J. Pennington, R. Socher, C.D. Manning, Glove: global vectors for word representation, in: A. Moschitti, B. Pang, W. Daelemans (Eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25–29, 2014, Doha, Qatar, A Meeting of SIGDAT, A Special Interest Group of the ACL, ACL, 2014, pp. 1532–1543..
[127] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: M.A. Walker, H. Ji, A. Stent (Eds.), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1–6, 2018, Volume 1 (Long Papers), Association for Computational Linguistics, 2018, pp. 2227–2237..
[128] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2–7,

2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186..
[129] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training, 2018..
[130] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, OpenAI Blog 1 (2019) 9.
[131] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D.M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, CoRR abs/2005.14165 (2020)..
[132] W. Liu, Z. Wang, N. Zeng, Y. Yuan, F.E. Alsaadi, X. Liu, A novel randomised particle swarm optimizer, Int. J. Mach. Learn. Cybern. 12 (2021) 529–540.
[133] N. Zeng, Z. Wang, W. Liu, H. Zhang, K. Hone, X. Liu, A dynamic neighborhood-based switching particle swarm optimization algorithm, IEEE Trans. Cybern. (2020).
[134] W. Liu, Z. Wang, Y. Yuan, N. Zeng, K. Hone, X. Liu, A novel sigmoid-function-based adaptive weighted particle swarm optimizer, IEEE Trans. Cybern. 51 (2021) 1085–1093.
[135] I.U. Rahman, Z. Wang, W. Liu, B. Ye, M. Zakarya, X. Liu, An n-state markovian jumping particle swarm optimization algorithm, IEEE Trans. Syst., Man, Cybern.: Syst. (2020).
[136] X. Luo, Y. Yuan, S. Chen, N. Zeng, Z. Wang, Position-transitional particle swarm optimization-incorporated latent factor analysis, IEEE Trans. Knowl. Data Eng. (2020).
[137] N. Zeng, D. Song, H. Li, Y. You, Y. Liu, F.E. Alsaadi, A competitive mechanism integrated multi-objective whale optimization algorithm with differential evolution, Neurocomputing 432 (2021) 170–182.
[138] J. Li, W. Monroe, D. Jurafsky, Understanding neural networks through representation erasure, CoRR abs/1612.08220 (2016)..
[139] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, I. Titov, Analyzing multi-head self-attention: specialized heads do the heavy lifting, the rest can be pruned, in: A. Korhonen, D.R. Traum, L. Màrquez (Eds.), Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28–August 2, 2019, Volume 1: Long Papers, Association for Computational Linguistics, 2019, pp. 5797–5808..
[140] Z. Dai, Z. Yang, Y. Yang, J.G. Carbonell, Q.V. Le, R. Salakhutdinov, Transformer-xl: Attentive language models beyond a fixed-length context, in: A. Korhonen, D.R. Traum, L. Màrquez (Eds.), Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28– August 2, 2019, Volume 1: Long Papers, Association for Computational Linguistics, 2019, pp. 2978–2988..
[141] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, L. Kaiser, Universal transformers, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6–9, 2019, OpenReview.net, 2019..
[142] Q. Guo, X. Qiu, P. Liu, Y. Shao, X. Xue, Z. Zhang, Star-transformer, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2–7, 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 1315–1325..
[143] X. Zhu, D. Cheng, Z. Zhang, S. Lin, J. Dai, An empirical study of spatial attention mechanisms in deep networks, in: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27–November 2, 2019, IEEE, 2019, pp. 6687–6696..
[144] Y. Tay, D. Bahri, D. Metzler, D. Juan, Z. Zhao, C. Zheng, Synthesizer: rethinking self-attention in transformer models, CoRR abs/2005.00743 (2020)..
[145] Y.H. Tsai, S. Bai, M. Yamada, L. Morency, R. Salakhutdinov, Transformer dissection: An unified understanding for transformer's attention via the lens of kernel, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019, Association for Computational Linguistics, 2019, pp. 4343–4352..
[146] A. Katharopoulos, A. Vyas, N. Pappas, F. Fleuret, Transformers are rnns: Fast autoregressive transformers with linear attention, CoRR abs/2006.16236 (2020).
[147] C. Sen, T. Hartvigsen, B. Yin, X. Kong, E.A. Rundensteiner, Human attention maps for text classification: Do humans and neural networks focus on the same words?, in: D. Jurafsky, J. Chai, N. Schluter, J.R. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020, Association for Computational Linguistics, 2020, pp. 4596–4608..

Zhaoyang Niu received his B.S. degree in the School of Data Science and Software Engineering from Qingdao University, Qingdao, China, in 2019. Now, he is studying for his BSc. degree in Computer Technology at the Ocean University of China, Qingdao, China. His research interests include computer vision, deep learning and attention mechanism.

Guoqiang Zhong received his B.S. degree in Mathematics from Hebei Normal University, Shijiazhuang, China, his M.S. degree in Operations Research and Cybernetics from Beijing University of Technology (BJUT), Beijing, China, and his Ph.D. degree in Pattern Recognition and Intelligent Systems from Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China, in 2004, 2007 and 2011, respectively. Between October 2011 and July 2013, he was a Post-doctoral Fellow with the Synchromedia Laboratory for Multimedia Communication in Telepresence, University of Quebec, Montreal, Canada. Between March 2014 and December 2020, he was an associate professor at Department of Computer Science and Technology, Ocean University of China, Qingdao, China. Since January 2021,he has been a full professor at Department of Computer Science and Technology, Ocean University of China. He has published 4 books, 4 book chapters and more than 80 technical papers in the areas of artificial intelligence, pattern recognition, machine learning and computer vision. His research interests include pattern recognition, machine learning and computer vision. He has served as Chair/PC member/reviewer for many international conferences and top journals, such as IEEE TNNLS, IEEE TKDE, IEEE TCSVT, Pattern Recognition, Knowledge-Based Systems, Neurocomputing, ACM TKDD, AAAI, AISTATS, ICPR, IJCNN, ICONIP and ICDAR. He has been awarded outstanding reviewer by several journals, such as Pattern Recognition, Knowledge-Based Systems, Neurocomputing and Cognitive Systems Research. He has won the Best Paper Award of BICS2019 and the APNNS Young Researcher Award. He is member of ACM, IEEE, IAPR, APNNS and CCF, professional committee member of CAAI-PR, CAA-PRMI and CSIG-DIAR, and trustee of Shandong Association of Artificial Intelligence.

Hui Yu is a Professor with the University of Portsmouth, UK. Prof. Yu received PhD from Brunel University London. He used to work at the University of Glasgow before moving to the University of Portsmouth. His research interests include methods and practical development in vision, machine learning and AI with applications to human-machine interaction, Virtual and Augmented reality, robotics and 4D facial expression. He serves as an Associate Editor of IEEE Transactions on Human-Machine Systems and Neurocomputing journal.