An Attentive Survey of Attention Models

SNEHA CHAUDHARI, VARUN MITHAL, GUNGOR POLATKAN, and
ROHAN RAMANATH, LinkedIn Corporation, USA

Attention Model has now become an important concept in neural networks that has been researched within diverse application domains. This survey provides a structured and comprehensive overview of the developments in modeling attention. In particular, we propose a taxonomy that groups existing techniques into coherent categories. We review salient neural architectures in which attention has been incorporated and discuss applications in which modeling attention has shown a significant impact. We also describe how attention has been used to improve the interpretability of neural networks. Finally, we discuss some future research directions in attention. We hope this survey will provide a succinct introduction to attention models and guide practitioners while developing approaches for their applications.

CCS Concepts: • **Computing methodologies** → **Neural networks**; *Natural language processing*; *Computer vision;*

Additional Key Words and Phrases: Attention, attention models, neural networks

ACM Reference format:
Sneha Chaudhari, Varun Mithal, Gungor Polatkan, and Rohan Ramanath. 2021. An Attentive Survey of Attention Models. *ACM Trans. Intell. Syst. Technol.* 12, 5, Article 53 (October 2021), 32 pages.
https://doi.org/10.1145/3465055

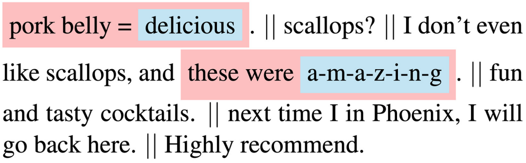
1 INTRODUCTION

Attention Model (AM), first introduced for Machine Translation [Bahdanau et al. 2015] has now become a predominant concept in neural network literature. Attention has become enormously popular within the **Artificial Intelligence (AI)** community as an essential component of neural architectures for a remarkably large number of applications in **Natural Language Processing (NLP)** [Galassi et al. 2020], Speech [Cho et al. 2015], and **Computer Vision (CV)** [Wang and Tax 2016].

The intuition behind attention can be best explained using human biological systems. For example, our visual processing system tends to focus selectively on some parts of the image, while ignoring other irrelevant information in a manner that can assist in perception [Xu et al. 2015]. Similarly, in several problems involving language, speech or vision, some parts of the input are more important than others. For instance, in machine translation and summarization tasks, only certain words in the input sequence may be relevant for predicting the next word. Likewise, in

Authors' address: S. Chaudhari, V. Mithal, G. Polatkan, and R. Ramanath, LinkedIn Corporation, 700 E. Middlefield Rd, Mountain View, California, 94043; emails: snchaudhari@linkedin.com, chaudharissneha@gmail.com, vamithal@linkedin.com.

2157-6904/2021/10-ART53 $15.00
https://doi.org/10.1145/3465055



pork belly = delicious . || scallops? || I don't even like scallops, and these were a-m-a-z-i-n-g . || fun and tasty cocktails. || next time I in Phoenix, I will go back here. || Highly recommend.

Fig. 1. Example of attention modeling in sentiment classification of Yelp reviews. Figure from Yang et al. [2016].

an image captioning problem, some regions of the input image may be more relevant for generating the next word in the caption. Attention Model incorporates this notion of relevance by allowing the model to dynamically *pay attention to* only certain parts of the input that help in performing the task at hand effectively. An example of sentiment classification of Yelp reviews [Yang et al. 2016] using AM is shown in Figure 1. In this example, the AM learns that out of five sentences, the first and third sentences are more relevant. Furthermore, the words *delicious* and *amazing* within those sentences are more meaningful to determine the sentiment of the review.

The rapid advancement in modeling attention in neural networks is primarily due to three reasons. First, these models are now the state-of-the-art for multiple tasks in NLP (Machine Translation, Summarization, Sentiment Analysis, and Part-of-Speech tagging) [Galassi et al. 2020], Computer Vision (Image Classification, Object Detection, Image Generation) [Khan et al. 2021], cross-modal tasks (Multimedia Description, Visual Question Answering) [Young et al. 2018], and Recommender Systems [Zhang et al. 2019b]. Second, they offer several other advantages beyond improving performance on the main task. They have been extensively used for improving interpretability of neural networks, which are otherwise considered as black-box models. This is a notable benefit, mainly because of growing interest in the fairness, accountability, and transparency of Machine Learning models in applications that influence human lives. Third, they help overcome some challenges with **Recurrent Neural Networks (RNNs)** such as performance degradation with increase in length of the input and the computational inefficiencies resulting from sequential processing of input (Section 3).

Organization: In this work, we aim to provide a brief, yet comprehensive survey on attention modeling. In Section 2, we build the intuition for the concept of attention using a simple regression model. We briefly explain the AM proposed by Bahdanau et al. [2015] and other attention functions in Section 3 and describe our taxonomy in Section 4. We then discuss key neural architectures using AM and present applications where attention has been widely applied in Section 5 and 6 respectively. Finally, we describe how attention is facilitating the interpretability of neural networks in Section 7 and conclude the article with future research directions in Section 8.

Related surveys: There have been a few domain-specific surveys on attention focusing on Computer Vision [Wang and Tax 2016], graphs [Lee et al. 2019], and Natural Language Processing [Galassi et al. 2020]. However, we further incorporate an accessible taxonomy, key architectures and applications, and interpretability aspect of AM. We hope that our contributions will not only foster broader understanding of AM but also help AI developers and engineers to determine the right approach for their application domain.

2 ATTENTION BASICS

The idea of attention can be understood using a regression model proposed by Naradaya–Watson in 1964 [Nadaraya 1964; Watson 1964]. We are given a training data of n instances comprising features and their corresponding target values $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. We want to predict the target value $\hat{y}$ for a new query instance x. A naive estimator will predict the simple average

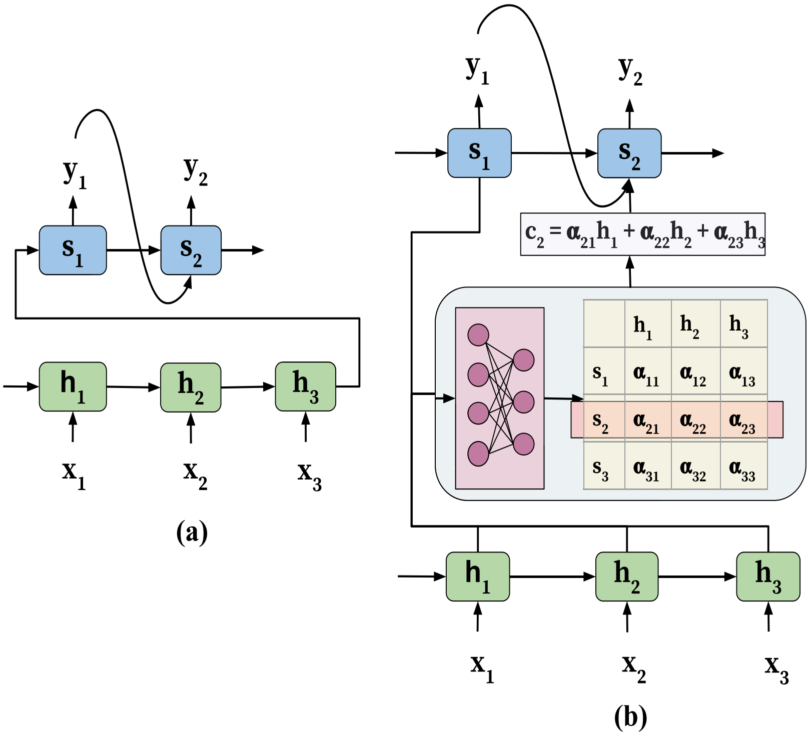

y_1
y_2
s_1
s_2
h_1
h_2
h_3
x_1
x_2
x_3
(a)
y_1
y_2
s_1
s_2
$c_2 = \alpha_{21}h_1 + \alpha_{22}h_2 + \alpha_{23}h_3$
h_1 h_2 h_3
s_1 α_{11} α_{12} α_{13}
s_2 α_{21} α_{22} α_{23}
s_3 α_{31} α_{32} α_{33}
h_1
h_2
h_3
x_1
x_2
x_3
(b)

Fig. 2. Encoder-decoder architecture: (a) traditional (b) with attention model.

of target values of all training instances: $\hat{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. Naradaya–Watson proposed a better approach in which the estimator uses a weighted average where weights correspond to relevance of the training instance to the query: $\hat{y} = \sum_{i=1}^{n} \alpha(x, x_i) y_i$. Here weighting function $\alpha(x, x_i)$ *encodes the relevance of instance x_i to predict for x*. A common choice for the weighting function is a normalized Gaussian kernel, though other similarity measures can also be used with normalization. The authors showed that the estimator has (i) consistency (given enough training data it converges to optimal results) and (ii) simplicity (no free parameters, the information is in the data and not in the weights). Fast-forward 50 years, attention mechanism in deep models can be viewed as a generalization that also allows learning the weighting function.

3 ATTENTION MODEL

The first use of AM was proposed by Bahdanau et al. [2015] for a sequence-to-sequence modeling task. A sequence-to-sequence model consists of an encoder-decoder architecture [Cho et al. 2014b] as shown in Figure 2(a). The encoder is an RNN that takes an input sequence of tokens $\{x_1, x_2, \ldots, x_T\}$, where T is the length of input sequence, and encodes it into fixed length vectors $\{h_1, h_2, \ldots, h_T\}$. The decoder is also an RNN that then takes a single fixed length vector h_T as its input and generates an output sequence $\{y_1, y_2, \ldots, y_{T'}\}$ token by token, where T' is the length of output sequence. At each position t, h_t and s_t denote the hidden states of the encoder and decoder respectively.

Challenges of traditional encoder-decoder: There are two well known challenges with this traditional encoder-decoder framework. First, the encoder has to compress all the input information into a single fixed length vector h_T that is passed to the decoder. Using a single fixed

Table 1. Encoder-decoder Architecture: Traditional and with Attention Model

Function	Traditional Encoder-Decoder	Encoder-Decoder with Attention
Encode	$h_i = f(x_i, h_{i-1})$	$h_i = f(x_i, h_{i-1})$
Context	$c = h_T$	$c_j = \sum_{i=1}^{T} \alpha_{ij} h_i$ $\alpha_{ij} = p(e_{ij})$ $e_{ij} = a(s_{j-1}, h_i)$
Decode	$s_j = f(s_{j-1}, y_{j-1}, c)$	$s_j = f(s_{j-1}, y_{j-1}, c_j)$
Generate	$y_j = g(y_{j-1}, s_j, c)$	$y_j = g(y_{j-1}, s_j, c_j)$

$x = (x_1, \ldots, x_T)$: input sequence, T: length of input sequence, h_i: hidden states of encoder, c: context vector, α_{ij}: attention weights over input, s_j: decoder hidden state, y_j: output token, f, g: non-linear functions, a: alignment function, p: distribution function.

length vector to compress long and detailed input sequences may lead to loss of information [Cho et al. 2014a]. Second, it is unable to model alignment between input and output sequences, which is an essential aspect of structured output tasks such as translation or summarization [Young et al. 2018]. Intuitively, in sequence-to-sequence tasks, each output token is expected to be more influenced by some specific parts of the input sequence. However, decoder lacks any mechanism to selectively focus on relevant input tokens while generating each output token.

Key idea: AM aims at mitigating these challenges by allowing the decoder to access the entire encoded input sequence $\{h_1, h_2, \ldots, h_T\}$. The central idea is to induce attention weights α over the input sequence to prioritize the set of positions where relevant information is present for generating the next output token.

Usage of attention: The corresponding encoder-decoder architecture with attention is shown in Figure 2(b). The attention block in the architecture is responsible for automatically learning the attention weights α_{ij}, which capture the relevance between h_i (the encoder hidden state) and s_{j-1} (the decoder hidden state). Note that the query state s_{j-1} is hidden state of the decoder just before emitting s_j and y_j. These attention weights are then used for building a context vector c, which is passed as an input to the decoder. At each decoding position j, the context vector c_j is a weighted sum of all hidden states of the encoder and their corresponding attention weights, i.e., $c_j = \sum_{i=1}^{T} \alpha_{ij} h_i$. This additional context vector is the mechanism by which decoder can access the entire input sequence and also focus on the relevant positions in the input sequence. This not only leads to improvements in performance on the final task but also improves the quality of the output due to better alignment. The same concept is shown mathematically in Table 1. The only major difference in the encoder-decoder architecture with attention is the composition of context vector c. In the traditional framework, context vector is just the last hidden state of the encoder h_T. In the attention-based framework, context at a given decoding step j is combination of all hidden states of the encoder and their corresponding attention weights; $c_j = \sum_{i=1}^{T} \alpha_{ij} h_i$.

Learning attention weights: The attention weights are learned by incorporating an additional feed-forward neural network within the architecture. This feed-forward network learns a particular attention weight α_{ij} as a function of two states, h_i (encoder hidden state) and s_{j-1} (decoder hidden state) that are taken as input by the neural network. This function is called the alignment function (denoted by a in Table 1) as it scores how relevant is the encoder hidden state h_i for the decoder hidden state s_{j-1}. This alignment function outputs energy scores e_{ij} that are then fed into the distribution function (denoted by p in Table 1) that converts the energy scores into attention weights. When the functions a and p are differentiable, the whole attention-based encoder-decoder

Table 2. Sumary of Alignment Functions

Function	Equation	References
similarity	$a(k_i, q) = sim(k_i, q)$	Graves et al. [2014a]
dot product	$a(k_i, q) = q^T k_i$	Luong et al. [2015a]
scaled dot product	$a(k_i, q) = \frac{q^T k_i}{\sqrt{d_k}}$	Vaswani et al. [2017]
general	$a(k_i, q) = q^T W k_i$	Luong et al. [2015a]
biased general	$a(k_i, q) = k_i(Wq + b)$	Sordoni et al. [2016]
activated general	$a(k_i, q) = act(q^T W k_i + b)$	Ma et al. [2017b]
generalized kernel	$a(k_i, q) = \phi(q)^T \phi(k_i)$	Choromanski et al. [2021]
concat	$a(k_i, q) = w_{imp}^T act(W[q; k_i] + b)$	Luong et al. [2015a]
additive	$a(k_i, q) = w_{imp}^T act(W_1 q + W_2 k_i + b)$	Bahdanau et al. [2015]
deep	$a(k_i, q) = w_{imp}^T E^{(L-1)} + b^L$ $E^{(l)} = act(W_l E^{(l-1)} + b^l$ $E^{(1)} = act(W_1 k_i + W_0 q) + b^l$	Pavlopoulos et al. [2017]
location-based	$a(k_i, q) = a(q)$	Luong et al. [2015a]
feature-based	$a(k_i, q) = w_{imp}^T act(W_1 \phi_1(K) + W_2 \phi_2(K) + b)$	Li et al. [2019a]

$a(k_i, q)$: alignment function for query q and key k_i, sim: similarity functions such as cosine, d_k: length of input, $(W, w_{imp}, W_0, W_1, W_2)$: trainable parameters, b: trainable bias term, act: activation function.

model becomes one large differentiable function and can be trained jointly with encoder-decoder components of the architecture using simple backpropagation.

Generalized Attention Model: The attention model shown in Figure 2(b) can also be seen as a mapping of sequence of keys K to an attention distribution α according to query q where keys are encoder hidden states h_i and query is the single decoder hidden state s_{j-1}. Here the attention distribution α_{ij} emphasizes the keys that are relevant for the main task with respect to the query q. Then $e = a(K, q)$ and $\alpha = p(e)$. In some cases, there is also additional input of values V on which the attention distribution is applied. The keys and values generally have one to one mapping and although the core attention model proposed by Bahdanau et al. [2015] does not distinguish between keys and values ($k_i = v_i = h_i$), some existing literature uses this terminology for different representations of the same input data. Hence a generalized attention model A works with a set of key-value pairs (K, V) and query q such that:

$$A(q, K, V) = \sum_i p(a(k_i, q)) * v_i. \tag{1}$$

As a concrete example, one can look at the regression task estimator explained in Section 2. Here the instance x is the query, the training data points x_i are keys and their labels y_i are values.

The alignment function (denoted by a) and distribution function (denoted by p) determine how keys and query are combined to produce attention weights. We discuss some of the commonly used alignment functions and distribution functions in the literature; we refer the reader to Galassi et al. [2020] for a more detailed discussion on alignment and distribution functions.

Alignment functions: The first major category of alignment functions are based on a notion of comparing query representations with key representations. For example, one approach is to compute either the *cosine similarity* or the *dot product* between the key and query representations (see Table 2). To account for varying lengths of representation, *scaled dot product* can be employed that normalizes the dot product by the representation vector length. Note that these functions assume that key and query have the same representation vector space. *General alignment* extends dot product to keys and queries with different representations by introducing a learnable transformation matrix **W** that maps queries to the vector space of keys. *Biased general alignment* allows to learn the global importance of some keys irrespective of the query by introducing a bias term. *Activated general alignment* adds a nonlinear activation layer such as hyperbolic tangent, rectifier linear unit, or scaled exponential linear unit. More recently, Choromanski et al. [2021] show that key and query can be matched using a generalized kernel function instead of the more commonly used *dot product*. The formulations of these alignment functions are presented in the Table 2.

The second major category of alignment functions combine keys and query to form a joint representation. One of the simplest models that follow this approach is the *concat alignment* by Luong et al. [2015a], where a joint representation is given by concatenating keys and queries. *Additive alignment* reduces computational time by decoupling the contributions of the query and the key; this allows precomputing contributions of all keys to avoid re-computation for each query. In contrast to a single neural layer used in additive alignment, *deep alignment* employs multiple neural layers.

There are some alignment functions that are designed for specific use-cases. *Location-based alignment* ignores the keys and only depends on q. The alignment score associated with each key is thus computed as a function of the key's position, independently of its content. Li et al. [2019a] show that when working with group of items such as two-dimensional (2D) patches for images or 1D temporal sequences, *derived features* (such as mean and standard deviation) from the representations of the individual elements belonging to the group can be used as input to alignment functions (such as additive alignment used in the article).

Distribution functions: Distribution functions map alignment function scores to attention weights. The most commonly used distribution functions are *logistic sigmoid* and *softmax*. These functions ensure that attention weights are constrained in [0,1] and sum to 1. Such weights can thus be interpreted as probabilities that an element is relevant. In case of softmax function, attention weights can be interpreted as the probability that the corresponding element is the most relevant. Most variants employ a softmax transformation in their attention mechanism, leading to dense alignments. This density is wasteful, making models less interpretable and assigning probability mass to many implausible outputs. Distribution functions such as *sparsemax* [Martins and Astudillo 2016] and *sparse entmax* [Martins et al. 2020; Peters et al. 2019] are able to produce sparse alignments and assign nonzero probability to only a short list of plausible outputs. Sparse distributions could be especially useful in applications such as document summarization or question-answering tasks where a large number of elements are irrelevant. Finally, compositional de-attention networks [Tay et al. 2019] introduced a distribution function that forms quasi-attention by using elementwise multiplication of two terms: $tanh(\frac{qk_i^T}{\sqrt{d_k}})$ and $sigmoid(\frac{G(qk_i^T)}{\sqrt{d_k}})$ (where G(.) is the negation of outer L1 distance between q against all keys). In this case, the first term controls the adding and subtracting of vectors. This is in contrast to traditional attention that only adds (weighted) vectors. The secondary term can be interpreted as a type of gating mechanism that deletes tokens that are irrelevant for the query (by making their contribution zero).

In this section, we discussed the seminal model that proposed attention mechanism for a sequence-to-sequence task in an encoder-decoder architecture. While the core idea remains the same, several extensions of attention modeling have been proposed in the literature to solve specific problem formulations. We also discussed different alignment and distribution functions used in literature. Next, we will see how attention formulations can be considerably different from each other in (i) the type of attention mechanism being used, (ii) the neural architectures, and (iii) the application domains. Figure 3 shows the three key components of any attention modeling technique.

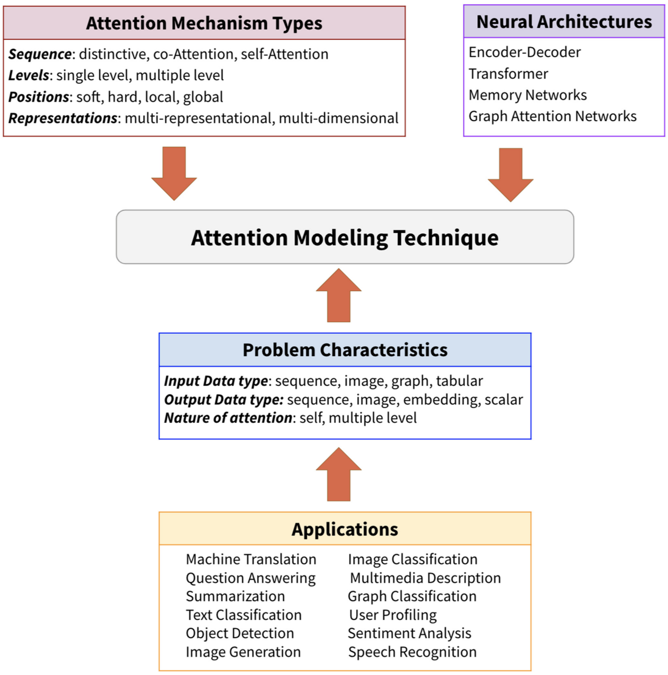

Attention Mechanism Types
Sequence: distinctive, co-Attention, self-Attention
Levels: single level, multiple level
Positions: soft, hard, local, global
Representations: multi-representational, multi-dimensional
Neural Architectures
Encoder-Decoder
Transformer
Memory Networks
Graph Attention Networks
Attention Modeling Technique
Problem Characteristics
Input Data type: sequence, image, graph, tabular
Output Data type: sequence, image, embedding, scalar
Nature of attention: self, multiple level
Applications
Machine Translation
Question Answering
Summarization
Text Classification
Object Detection
Image Generation
Image Classification
Multimedia Description
Graph Classification
User Profiling
Sentiment Analysis
Speech Recognition

Fig. 3. Key components of Attention Modeling Techniques.

In the remainder of this survey, we will discuss a taxonomy of attention types, key neural architectures using AM and their differences, and how AM has been applied to some applications.

4 TAXONOMY OF ATTENTION

We consider attention in four broad categories and elucidate the different types of attention within each category as shown in Table 3. Note that these categories are not mutually exclusive. In fact, one can think of these categories as dimensions along which attention can be considered while employing it for an application of interest. For example, a multi-level, self- and soft attention combination has been used by Yang et al. [2016]. To make this concept comprehensible, we provide a list of key technical papers and specify the multiple types of attention used within the proposed approaches in Table 4.

4.1 Number of Sequences

Thus far, we have only considered the case that involves a single input and corresponding output sequence. This type of attention, which we refer to as *distinctive*, is used when key and query states belong to two distinct input and output sequences respectively. Most attention models employed for translation [Bahdanau et al. 2015], image captioning [Xu et al. 2015], and speech recognition [Chan et al. 2016] fall within the distinctive type of attention.

A *co-attention* model operates on multiple input sequences at the same time and jointly learns their attention weights, to capture interactions between these inputs. Lu et al. [2016] used a co-attention model for visual question answering. The authors argued that in addition to modeling visual attention on the input image, it is also important to model question attention, because all words in the text of question are not equally important to the answer of the question. Further, attention-based image representation is used to guide the question attention and vice versa, which

Table 3. Table Presents Key Characteristics of Different Types of Attention within Each Category

Category	Type	Characteristics
# Sequences	distinctive	* Key and query states are from two distinct sequences.
	co-attention	* Multiple input sequences. * Very useful for modeling multi-modal data.
	self	* Key and query states are from the same sequence. * Most popular form of attention in recent papers.
# Abstractions	single-level	* Attention is computed only once on the original input.
	multi-level	* Hierarchical attention on multiple abstractions of input. * Better performance as natural hierarchy in text/images. * Computational cost increases with number of levels.
# Positions	soft/global	* Weighting over all positions. Computationally intensive. * Nice differentiable training objective.
	hard	* Picks some positions by sampling from predictor. * Computationally more efficient than soft. * Lacks differentiability of training objective.
	local	* Soft-attention in a window around a position. * Offers tradeoff: efficient and locally differentiable.
# Representations	multi-representational	* Deals with multiple representations of the same input. * Selects representations relevant for downstream tasks.
	multi-dimensional	* Computes relevance over each dimension of input. * Extracts contextual meaning of input dimensions.

essentially helps to simultaneously detect key phrases in the question and corresponding regions of images relevant to the answer. Similarly, Yu et al. [2019] use co-attention for visual question answering task.

In contrast, for tasks such as text classification and recommendation, input is a sequence but the output is not a sequence. In this scenario, attention can be used for learning relevant tokens in the input sequence for every token in the *same* input sequence. In other words, the key and query states belong to the same sequence for this type of attention. For this purpose, *self*-attention, also known as inner attention has been proposed by Yang et al. [2016]. To understand this better, let's consider an input sequence of words $\{w_1, w_2, w_3, w_4, w_5\}$ such that w_i is the vector representation of the words in the sequence. If we feed this input sequence to a self-attention layer, then the output is another sequence $\{y_1, y_2, y_3, y_4, y_5\}$ such that $y_i = \sum_j \alpha_{ij} * w_j$. Here the attention weights aim to capture how two words in the same sequence are related, where the concept of relevance depends on the main task.

4.2 Number of Abstraction Levels

In the most general case, attention weights are computed only for the original input sequence. This type of attention can be termed as *single-level.* However, attention may be applied on multiple levels of abstraction of the input sequence in a *sequential* manner. The output (context vector) of the lower abstraction level becomes the query state for the higher abstraction level. Additionally, models that use *multi-level* attention can be further classified based on whether the weights are learned top-down [Zhao and Zhang 2018] (from higher level of abstraction to lower level) or bottom-up [Yang et al. 2016].

We illustrate a key example in this category that uses the attention model at two different levels of abstraction, i.e., at word level and sentence level, for the document classification task [Yang et al. 2016]. This model is called a **Hierarchical Attention Model (HAM)**, because it captures the natural hierarchical structure of documents, i.e., a document is made up of sentences

Table 4. Summary of Key Papers for Technical Approaches in AMs

Reference	Application	Category			
		Number of Sequences	Number of Abstraction Levels	Number of Representations	Number of Positions
Bahdanau et al. [2015]	Machine Translation	distinctive	single-level	—	soft
Xu et al. [2015]	Image Captioning	distinctive	single-level	—	hard
Luong et al. [2015b]	Machine Translation	distinctive	single-level	—	local
Yang et al. [2016]	Document Classification	self	multi-level	—	soft
Chan et al. [2016]	Speech Recognition	distinctive	single-level	—	soft
Lu et al. [2016]	Visual Question Answering	co-attention	multi-level	—	soft
Wang et al. [2017]	Sentiment Classification	co-attention	multi-level	—	soft
Ying et al. [2018a]	Recommender Systems	self	multi-level	—	soft
Shen et al. [2018]	Language Understanding	self	single-level	multi-dimensional	soft
Kiela et al. [2018]	Text Representation	self	single-level	multi-representational	soft

"—" means not applicable.

and sentences are made up of words. The multi-level attention allows the HAM to extract words that are important in a sentence and sentences that are important in a document as follows. It first builds an attention-based representation of sentences with first level attention applied on sequence of word embedding vectors. Then it aggregates these sentence representations using a second level attention to form a representation of the document. This final representation of the document is used as a feature vector for the classification task.

Stacked Attention Networks (SANs) proposed in Sun and Fu [2019] also fall into this category as they mainly employ multiple layers to iteratively refine the attention by combining information from the query (question) and results of previous attention layers. For example, the authors in Sun and Fu [2019] used SANs for image question answering task where multiple attention layers query the image multiple times to progressively to locate the exact regions in the image that are highly relevant for the answer. Authors claim that using global image presentation to predict the answer leads to sub-optimal results, as the attention is scattered on many objects within the first layer. But when multiple attention layers are used, higher level attention layers utilize the knowledge from lower level attention layers (visual information) and the refined query vector (question information) to extract more fine-grained and smaller regions within the image. They also observed that two attention layers are better than one, but three or more layers did not further improve the performance.

Note that the co-attention work [Lu et al. 2016] described in Section 4.1 also belongs to multi-level category where it co-attends to the image and question at three levels: word level, phrase level and question level. This combination of co-attention and multi-level attention is depicted in Figure 4. Zhao and Zhang [2018] proposed "attention-via-attention," which uses multi-level

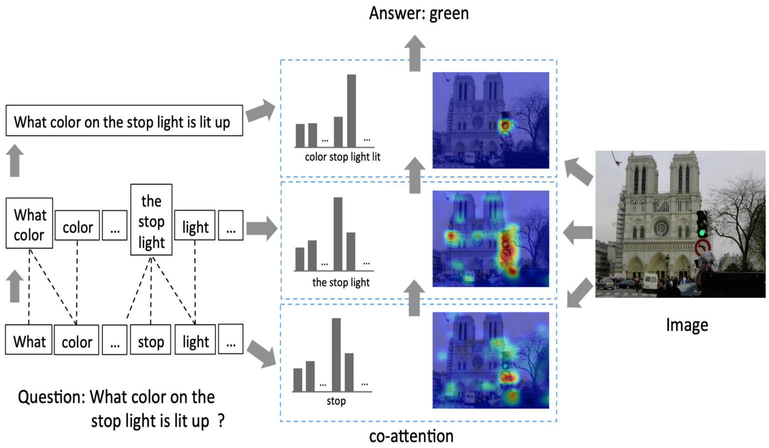


Fig. 4. The AM proposed by Lu et al. [2016] for Visual Question Answering task that is a combination of co-attention (visual and text) and multi-level (word level, phrase level and question level) attention.

attention (with characters on the lower level and words on the higher level) and learns the attention weights in top-down fashion.

4.3 Number of Positions

In the third category, the differences arise from positions of the input sequence where attention function is calculated. The attention introduced by Bahdanau et al. [2015] is also known as *soft* attention. As the name suggests, it uses a weighted average of all hidden states of the input sequence to build the context vector. The usage of the soft weighing method makes the neural network amenable to efficient learning through backpropagation, but also results in quadratic computational cost.

Xu et al. [2015] proposed a *hard* attention model in which the context vector is computed from stochastically sampled hidden states in the input sequence. This is accomplished using a multinoulli distribution parameterized by the attention weights. The hard attention model is beneficial due to decreased computational cost, but making a hard decision at every position of the input renders the resulting framework non-differentiable and difficult to optimize. Note that these categories are not mutually exclusive. Variational learning methods and policy gradient methods in reinforcement learning have been proposed in the literature to overcome this limitation.

Luong et al. [2015b] proposed two attention models, namely *local* and *global*, in context of machine translation task. The global attention model is similar to the soft attention model. The local attention model, however, is an intermediate between soft and hard attention. The key idea is to first detect an attention point or position within the input sequence and pick a window around that position to create a local soft attention model. The position within input sequence can either be set (monotonic alignment) or learned by a predictive function (predictive alignment). Consequently, the advantage of local attention is to provide a parametric tradeoff between soft and hard attention, computational efficiency and differentiability within the window.

4.4 Number of Representations

Generally a single feature representation of the input sequence is used in most applications. However, in some scenarios, using a single feature representation of the input may not suffice for the downstream task. In these cases, one approach is to capture different aspects of the input through multiple feature representations. Attention can be used to assign importance weights to these different representations, which can determine the most relevant aspects, disregarding noise and redundancies in the input. We refer to this model as *multi-representational AM*, as it can determine the relevance of multiple representations of the input for downstream application. The final representation is a weighted combination of these multiple representations and their attention weights. One benefit of attention here is to directly evaluate which embeddings are preferred for which specific downstream tasks by inspecting the weights.

Kiela et al. [2018] trained attention weights over different word embeddings of the same input sentence to improve sentence representations. Similarly, Maharjan et al. [2018] used attention to dynamically weigh different feature representations of books capturing lexical, syntactic, visual and genre information.

Based on similar intuition, in *multi-dimensional* attention, weights are induced for determining the relevance of each dimension of the input embedding vector. The intuition is that computing a score for each feature of the vector can select the features that can best describe the token's specific meaning in any given context. This is especially useful for natural language applications where word embeddings suffer from the polysemy problem. Examples of this approach are shown in Lin et al. [2017] for more effective sentence embedding representation and in Shen et al. [2018] for language understanding problem.

5 NETWORK ARCHITECTURES WITH ATTENTION

In this section, we describe some salient neural architectures used in conjunction with attention: (1) the Encoder-Decoder framework, (2) the Transformer that circumvents the sequential processing component of recurrent models with the use of attention, (3) Memory Networks that extend attention beyond a single input sequence, and (4) **Graph Attention Networks (GAT)**. These are some neural architectures that use AM extensively and have become popular choice in many application domains. However, exploring use of AM within various neural architectures is an active research topic, and the list of neural architectures using AM is growing fast.

5.1 Encoder-Decoder

The earliest use of attention was as part of RNN-based encoder-decoder framework to encode long input sentences [Bahdanau et al. 2015]. Consequently, attention has been most widely used with this architecture. An interesting fact is that AM can take any input representation and reduce it to a single fixed length context vector to be used in the decoding step. Thus, it allows one to decouple the input representation from the output. One could exploit this benefit to introduce hybrid encoder-decoders, the most popular being **Convolutional Neural Network (CNN)** as an encoder, and RNN or **Long Short Term Memory (LSTM)** as the decoder. This type of architecture is particularly useful for many multi-modal tasks such as Image and Video Captioning, Visual Question Answering and Speech Recognition.

However, not all problems where both input and output are sequential can be solved with the aforementioned formulation (e.g., sorting or travelling salesman problem). *Pointer networks* [Vinyals et al. 2015] are another class of neural models with the following two differences: (i) the output is discrete and points to positions in the input sequence (hence the name pointer network), and (ii) the number of target classes at each step of the output depends on the length of the input (and hence variable). This cannot be achieved using the traditional encoder-decoder framework where the output dictionary is known *a priori* (e.g., in case of natural language modeling). The authors achieved this using attention weights to model the probability of choosing the ith input symbol as the selected symbol at each output position. This approach can be applied to discrete optimization problems such as travelling salesperson problem and sorting [Vinyals et al. 2015].

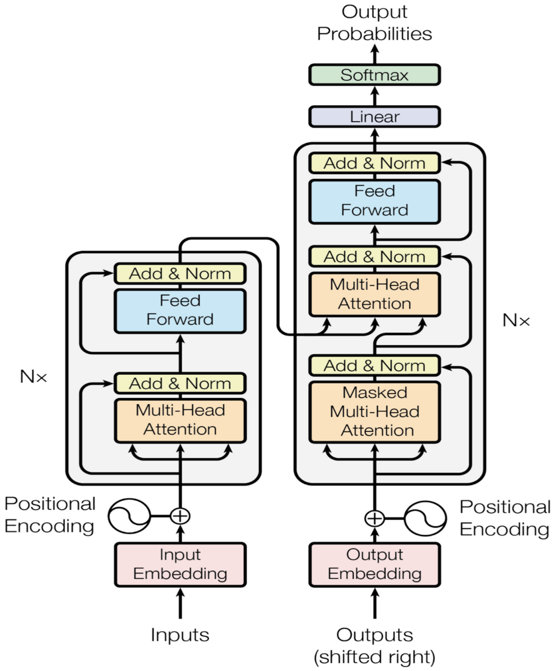


Fig. 5. Transformer Architecture. Figure from Vaswani et al. [2017].

5.2 Transformer

Recurrent architectures rely on sequential processing of input at the encoding step that results in computational inefficiency, as the processing cannot be parallelized [Vaswani et al. 2017]. To address this, the authors in Vaswani et al. [2017] proposed *Transformer* architecture that completely eliminates sequential processing and recurrent connections. It relies *only* on self-attention mechanism to capture global dependencies between input and output. Authors demonstrated that Transformer architecture achieved significant parallel processing, shorter training time and higher accuracy for Machine Translation without any recurrent component.

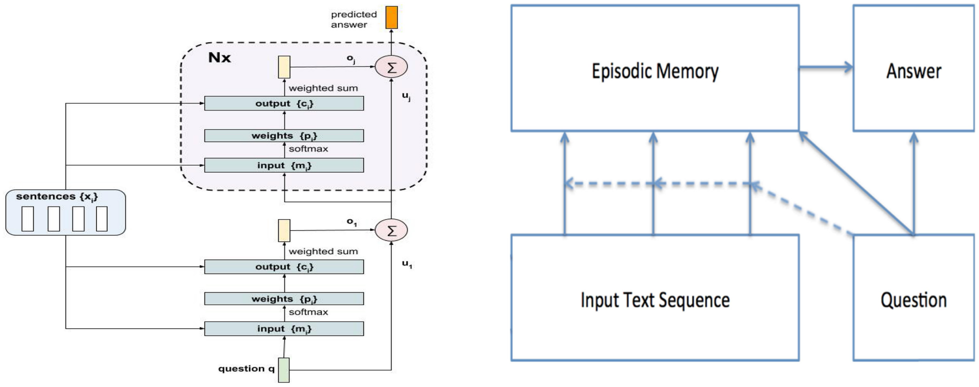
The Transformer architecture is shown in Figure 5. Transformer mainly relies on self-attention mechanism that relates tokens and their positions within the same input sequence. Authors propose a novel scaled dot product alignment function for self-attention mechanism, also explained in Section 3. Further, attention is known as *multi-head*, because several attention layers are stacked in parallel, with different linear transformations of the same input. In other words, rather than only computing the attention once, the multi-head mechanism splits the input into fixed-size segments and then computes the scaled dot-product attention over each segment in parallel. The independent attention outputs are then concatenated into expected dimensions. The main architecture is composed of a stack of six identical layers of encoders and decoders with two sub-layers: pointwise **Feed Forward Network (FFN)** layer and multi-head self-attention layer. Pointwise FFN means same linear transformation is applied to each position in the sequence independently, increasing parallel processing. The decoder is similar to the encoder, except that the decoder contains two multi-head attention sub-modules instead of one. The first multi-head attention sub-module is masked to prevent positions from attending to the future. One additional feature of this architecture is usage of positional encoding. The positional encoding is used, because input is sequential, which demands the model to make use of the temporal aspect of the input (order information), yet components that capture this positional information (i.e., RNNs/CNNs) are not used. To account for this, the encoder phase in the Transformer generates content embedding as well as positional encoding for each token of the input sequence. Finally, normalization and residual connections are mechanisms used to help the model train faster and more accurately.

Transformers can capture global/long range dependencies between input and output, support parallel processing, require minimal inductive biases (prior knowledge), demonstrate scalability to large sequences and datasets, and allow domain-agnostic processing of multiple modalities (text, images, speech) using similar processing blocks. Consequently, Transformer architecture has become state-of-the-art approach for many mainstream NLP, Computer Vision and Cross-Modal tasks. Moreover, there has been an increasing interest in the use of attention models in a wide variety of applications after Transformer, making this architecture a significant milestone for attention. The number of variants of Transformer and its applications have grown to the extent that individual surveys have been published on this topic, e.g., Khan et al. [2021]. We describe some key variants of Transformer and its applications for NLP, Computer Vision and Cross-Modal tasks in Sections 6.1, 6.2, and 6.3, respectively.

However, although Transformer is undoubtedly a huge improvement over the RNN-based sequential models, it comes with its own share of limitations: (i) input text has to split into fixed number of segments resulting in context fragmentation, (ii) high parametric complexity that results in computational cost and resources, (iii) large training data requirements due to minimal inductive bias, and (iv) difficulty in interpreting what self-attention mechanism learns and what is the contribution of input tokens toward predictions. Consequently, multiple lines of work have been established to introduce improvements in Transformers.

One direction of research analyzes the multi-head self-attention in Transformers. Clark et al. [2019] perform analysis of Transformer-based (pre-trained) language model called **Bidirectional Encoder Representations (BERT)** to determine that particular heads correspond well to particular relations such as direct objects of verbs, determiners of nouns, objects of prepositions, and so on. Further, first layer consists of high-entropy heads that produce bag-of-vector-like representations by having broader attention span. Voita et al. [2019] evaluate the contribution made by attention heads in the encoder to the overall performance of the model and find that (i) vast majority of heads can be pruned without affecting performance, (ii) important heads have specialized and linguistically interpretable functions in the model. Kobayashi et al. [2020] perform norm-based analysis of transformed input vectors to reveal that contrary to previous studies, BERT pays poor attention to special tokens, and word alignment can be extracted from attention mechanisms of Transformer. Last, Michel et al. [2019] find that many attention layers can even be individually reduced to a single attention head without affecting performance, increasing its efficiency and propose a novel technique for pruning.

Another line of work aims to improving attention span is to make the context that can be used in self-attention longer, more efficient and flexible. Transformer-XL by Dai et al. [2019] is one such work that solves the context fragmentation problem by adopting new positional encoding and reusing hidden states between segments. Another work by Sukhbaatar et al. [2019a] propose a novel self-attention mechanism that can learn its optimal attention span. This allows to extend the context size used in Transformer, while maintaining the computational cost. Finally, since the computational cost of Transformer grows quadratically with sequence length. Hence, an important research direction is to reduce the computation time and memory consumption. Sparse Transformers [Child et al. 2019], Reformers [Kitaev et al. 2020], and Performers [Choromanski et al. 2021] are some recent approaches toward this end.



(a) End-to-End Memory Network Architecture

(b) Dynamic Memory Network Architecture Overview. Figure from [Kumar et al. 2016]

Fig. 6. Memory Network Architectures.

5.3 Memory Networks

Applications like **Question Answering (QA)** and chat bots require the ability to learn from information in a database of facts. The input to the network is a knowledge database and a query, where some facts are more relevant to the query than others. In this case, an external memory is needed to store the knowledge database of facts and attention is crucial to selectively focus on the relevant facts. It can be considered as a generalization of attention, wherein instead of modeling attention over a single sequence, it is used over a large database of sequences (facts). We consider three approaches in the literature that couple an external memory component with attention for Question Answering namely, End-to-End Memory Networks [Sukhbaatar et al. 2015], Dynamic Memory Networks [Kumar et al. 2016], and Neural Turing Machines [Graves et al. 2014b]. We can think of memory networks as generally having three components: (i) A process that "reads" raw database, and converts them into distributed representations. (ii) A list of feature vectors storing the output of the reader. This can be understood as a "memory" containing a sequence of facts, which can be retrieved later, not necessarily in the same order, without having to visit all of them. (iii) A process that "exploits" the content of the memory to sequentially perform a task, at each time step having the ability put attention on the content of one memory element (or a few, with a different weight).

End-to-End Memory Networks (MemN2N) derive its name from the fact that they enable end-to-end training via backpropagation compared to Memory Networks [Weston et al. 2014] and require less supervision, making them applicable to wider range of NLP tasks such as Question Answering and Language Modeling. It also allows multiple reads ("hops") over the long-term memory before generating output token that is crucial for good performance these tasks. The architecture shown in Figure 6(a) can be broken down into two main parts. First part of the architecture helps to find the relevant facts for the query from the knowledge database using inner product between query and each memory vector, followed by softmax operation, to find the best match. In the second part, final answer for the query is calculated using the context vector over relevant facts with the help of attention.

Dynamic Memory Networks (DMN) use an episodic memory module, which chooses which parts of the inputs to focus on through the attention mechanism and outputs a memory vector representation. It repeats this process iteratively by conditioning the attention over the question as well as the previous memory representation, which allows the module to (i) attend to different inputs during each iteration (ii) retrieve additional information, which was thought to be irrelevant in previous iterations. Figure 6(b) shows an overview of the architecture. Questions trigger gates that control whether certain input are given to the episodic memory module. The final state of the episodic memory (after multiple episodes/iterations) is the input to the answer module. Another work by Xiong et al. [2016] demonstrates the use of Dynamic Memory Networks to answer questions based on images. The input module extracts feature vectors from images using a CNN-based network that are then fed to the episodic memory module.

Neural Turing Machine (NTM) also uses a continuous (albeit smaller) memory representation along with a controller (typically feed-forward network or LSTM) that dictates read/write operations on the memory. The system is similar to a Turing Machine but is differentiable end-to-end, allowing it to be efficiently trained with gradient descent. It uses attention to access the memory selectively, constraining the read and write operation to interact with a small portion of the memory and making interactions with the memory highly sparse. However, the NTM memory uses both content and address-based access and is used to infer simple algorithms such as copying, sorting and associative recall from input and output examples.

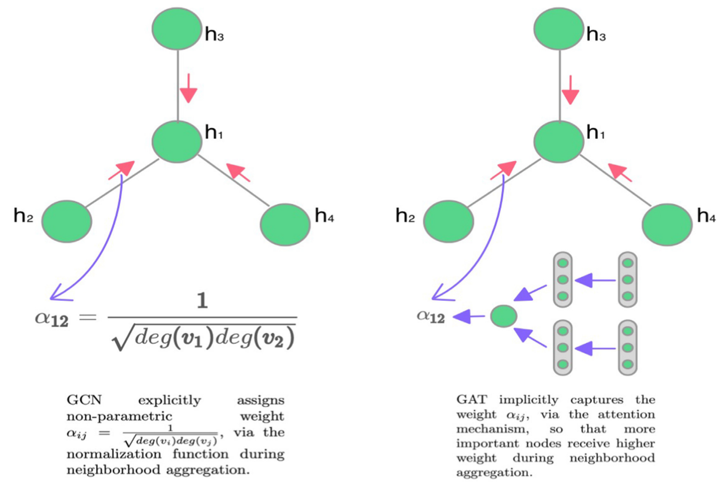
The biggest advantage of memory networks is that it can store information as memory and use it effectively and scalably via attention mechanism by focusing on only small, relevant part of the memory. MemN2N have shown superior performance on tasks such as Question Answering and Language Modeling compared to RNNs and LSTMs. DMNs have shown state-of-the-art results on Sentiment Analysis and Part of Speech Tagging.

5.4 Graph Attention Networks (GAT)

Application domains such as social networks, citation networks, protein-protein interactions, cheminformatics, and so on, come naturally in the form of graph-structured data. Arbitrary graphs have a varying number of neighbors and it is therefore considerably harder to work with them compared to sequences and images. In fact, one could think of sequences and images as special form of graphs with highly rigid and regular connectivity pattern (each element in sequence is connected to two adjacent elements; each pixel in an image is connected to its eight neighbouring pixels).

Generalizing convolutional layers used for images to graph convolutional layers would thus require innovations to enable: (i) computational and storage efficiency (model should not require more than O(V+E) time and memory), (ii) fixed number of parameters (the number of parameters should not depend on the input graph size else the model becomes unmanageable), (iii) localisation (the model should be able to work independently on a local neighbourhood of a node to enable parallel computation and scalability), (iv) ability to specify arbitrary importance to different neighbours so as to correctly incorporate relevance of a neighbor for each node and not treat all neighbors equally or based on some structural property such as node degree, and (v) applicability to arbitrary, unseen graph structures.

Graph Convolutional Network (GCN) showed good performance on node classification task by combining local graph structure and node-level features. However, as shown in the Figure 7, GCN assigns explicit non-parametric weight to neighbors based on their node degree. Velickovic et al. [2018] propose GATs that employ self-attention over the node features of neighbors so that more important nodes receive higher weight. GATs are computationally efficient as the operation of the self-attention layer can be parallelized. No eigendecomposition or computationally intensive matrix operations are required. Moreover, GATs allow for assigning different importance to nodes of a same neighborhood via attention weights, enabling a higher model capacity than GCNs. Finally, the attention mechanism is applied in a shared manner to all edges in the graph, and therefore it does not depend on upfront access to the global graph structure.



GCN explicitly assigns non-parametric weight $\alpha_{ij} = \frac{1}{\sqrt{deg(v_i)deg(v_j)}}$, via the normalization function during neighborhood aggregation.

GAT implicitly captures the weight α_{ij}, via the attention mechanism, so that more important nodes receive higher weight during neighborhood aggregation.

Fig. 7. Neighbor importance in Graph Convolutional Network vs. Graph Attention Network. Figure from https://dsgiitr.com/blogs/gat/.

The real-world graph usually comes with multi-types of nodes and edges, also widely known as heterogeneous graphs. Heterogeneous graphs contain more comprehensive information and richer semantics, therefore are more useful for data mining tasks. However, due to this complexity of heterogeneous graphs, homogeneous graph approaches cannot be directly applied to heterogeneous graphs. Wang et al. [2019b] extend graph attention for heterogeneous graphs using hierarchical attention. In particular, given the node features as input, a transformation matrix projects different types of node features into the same space. Then the node-level attention is able to learn the attention values between the nodes and their meta-path (connections defined on multiple nodes)-based neighbors, while the semantic-level attention aims to learn the attention values of different meta-paths for the specific task in the heterogeneous graph. Based on the learned attention values in terms of the two levels, the model can get the optimal combination of neighbors and multiple meta-paths in a hierarchical manner, which enables the learned node embeddings to better capture the complex structure and rich semantic information. The overall model is optimized via backpropagation in an end-to-end manner.

6 APPLICATIONS

Attention models have become an active area of research because of their intuition, versatility, and interpretability. Variants of attention models have been used to address unique characteristics of a diverse set of application domains. In some applications, attention models have shown a significant impact on the performance for the task at hand, whereas in others they have helped to learn better representations of entities such as documents, images and graphs. In some cases, attention has entirely transformed the field of application by becoming the most popular choice of technique. A few such examples are Machine Translation, pre-trained embeddings with BERT and Question Answering.

Given that the areas of application are very broad, in this work, we mainly discuss the need for attention models for each application domain, a few instances of applications within each domain and cover their seminal work in Table 5 that can become a starting point for further investigation. We describe attention modeling in the following application domains: (i) NLP, (ii) Computer Vision, (iii) Multi-Modal Tasks, (iv) Recommender Systems, and (v) Graphical Systems.

Table 5. Summary of Key Applications of AMs

Application Domain	Application	Seminal Works
Natural Language Processing	Machine Translation	Zhang et al. [2020], Tang et al. [2018], Vaswani et al. [2017], Britz et al. [2017], Bahdanau et al. [2015], Luong et al. [2015b]
	Summarization	Xu et al. [2020], Nallapati et al. [2016], Chopra et al. [2016], Rush et al. [2015]
	Text Classification and Representation	Kiela et al. [2018], Lin et al. [2017], Yang et al. [2016]
	Sentiment Analysis	Wang et al. [2020c], Wang et al. [2016], Ma et al. [2018], Tang et al. [2016], Ambartsoumian and Popowich [2018]
	Question Answering	Dehghani et al. [2019], Hermann et al. [2015], Sukhbaatar et al. [2015]
	Pre-trained Language Models	Lewis et al. [2020], Lan et al. [2020], Brown et al. [2020], Yang et al. [2019], Dai et al. [2019], Devlin et al. [2019]
Computer Vision	Image Classification	Dosovitskiy et al. [2021], Touvron et al. [2021], Zhao et al. [2020], Jetley et al. [2018], Mnih et al. [2014]
	Image Generation	Chen et al. [2020], Parmar et al. [2018], Gregor et al. [2015],
	Object Detection	Zhu et al. [2021], Carion et al. [2020], Lin et al. [2020], Ba et al. [2014]
	Image Synthesis	Zhang et al. [2019a]
Multi-Modal Tasks	Multimedia (Image, Video) Description	Wang et al. [2020b], Sun et al. [2019], Xu et al. [2015], Yao et al. [2015], Cho et al. [2015]
	Speech Recognition	Qin and Qu [2020], Chan et al. [2016], Chorowski et al. [2015]
	Visual Question Answering	Lu et al. [2019], Tan and Bansal [2019], Li et al. [2019b], Anderson et al. [2018], Lu et al. [2016],
	Human Communication Comprehension	Zadeh et al. [2018]
Recommender System	User Profiling	Qiannan et al. [2019], Shuai Yu [2019], He et al. [2018], Zhou et al. [2018], Kang and McAuley [2018], Ying et al. [2018b]
	Item/User Representations	Zhu et al. [2020], Deng et al. [2020], Wu et al. [2019]
	Exploit Auxiliary Information	Xiao et al. [2021], Wu et al. [2020], Wang et al. [2019a]
Graph-based Systems	Graph Classification	Lee et al. [2018], Ma et al. [2017a]
	Graph to Sequence Generation	Zheng et al. [2020], Beck et al. [2018]
	Node Classification	Cui et al. [2020], Velickovic et al. [2018], Abu-El-Haija et al. [2018], Feng et al. [2016]
	Hyperedge Detection	Zhang et al. [2020]

6.1 Natural Language Processing (NLP)

In the NLP domain, attention assists in focusing on the relevant parts of the input sequence, alignment of input and output sequences, and capturing long range dependencies for longer sequences. For instance, modeling attention in neural techniques for *Machine Translation* allows for better alignment of sentences in different languages, which is a crucial problem in MT. This automatic alignment of sentences in different languages helps to capture subject-verb-noun locations that differ from language to language. The advantage of the attention model also becomes more apparent while translating longer sentences [Bahdanau et al. 2015]. The longer the sentence, the harder it is to embed all the content and alignment information in the vanilla technique without attention. Several studies including Britz et al. [2017] and Tang et al. [2018] have shown performance improvements in Machine Translation using attention. Zhang et al. [2020] have presented GRU-gated attention model for Machine Translation. The GRU-gating mechanism is useful to avoid computation of similar context vectors at different decoding steps, allowing them to be more discriminatory in nature.

Question Answering problems have made use of attention to better understand questions by focusing on relevant parts of the question [Hermann et al. 2015] and store large amount of information using memory networks to help find answers [Sukhbaatar et al. 2015]. Another seminal work by Rush et al. [2015] made significant advancement in abstractive sentence *summarization* task by using soft attention mechanism. Such data driven approaches had proven to be challenging in the past for the task of summarization but the proposed method showed significant gains compared to several existing baselines. Xu et al. [2020] uses a Transformer-based model to enhance the copy mechanism in abstractive summarization. The self-attention in Transformer is used to guide the copy mechanism so that it can focus on important words in the source text that need to be extracted into the summary.

In the *Sentiment Analysis* task, self-attention helps to focus on the words that are important for determining the sentiment of input. A couple of approaches for aspect-based sentiment classification by Wang et al. [2020c], Wang et al. [2016], and Ma et al. [2018] incorporate additional knowledge of aspect related concepts into the model and use attention to appropriately weigh the concepts apart from the content itself. Sentiment Analysis application has also seen multiple architectures being used with attention such as Memory Networks [Tang et al. 2016] and Transformer [Ambartsoumian and Popowich 2018].

Other applications within the NLP domain that have employed attention models extensively include Text Classification, and Text Representation Learning. As mentioned earlier in Section 4, *Text Classification* and *Text Representation* problems mainly make use of self-attention to build more effective sentence or document representations/embeddings. Yang et al. [2016] use a multi-level self-attention, whereas Lin et al. [2017] propose a multi-dimensional and Kiela et al. [2018] propose a multi-representational self-attention model.

Last, many applications of NLP have been completely revolutionised with the advent of *Pre-Trained Language Models*. Pre-trained language models have proven extremely beneficial due to the following reasons: (i) they are trained on large corpus that can learn universal language representations capturing many facets of language such as long-term dependencies, hierarchical relations, and sentiment; (ii) they can be easily fine tuned for multiple downstream NLP tasks with significantly less labeled data, avoiding training a new model from scratch; (iii) they democratize development of NLP applications by allowing easier model building. Transformer-based pre-trained language models have been especially popular recently with three main types: (i) Transformer and its variants such as Transformer-XL [Dai et al. 2019], BART [Lewis et al. 2020], (ii) Bidirectional BERT [Devlin et al. 2019] and its variants such as RoBERTa [Liu et al. 2019], ALBERT [Lan et al. 2020] (iii) Generative Pre-trained Transformer (GPT) and its variants such as GPT-2 [Radford et al. 2018], GPT-3 [Brown et al. 2020]. XLNet combines BERT and Transformer-XL [Yang et al. 2019].

Although Transformer has been used for several NLP tasks, Transformer-XL understands context beyond the fixed-length limitation of Transformer and can learn 450% longer dependency, critical to achieve better performance on both short and long sequences. BART is similar to Transformer in architecture but is trained to reconstruct the original text from corrupted input text with an arbitrary noising function. However, BERT uses bi-directional encoder such that it allows the model to consider the context from both the left and the right sides of each word. RoBERTa improves over BERT by using a larger dataset for training, training the model over more iterations, and removing the next sequence prediction training objective. ALBERT uses parameter reduction techniques such as factorized embedding parameterization, cross-layer parameter sharing to reduce the number of parameters by 18× and faster training by 1.7×. Finally XLNet combines the capabilities of BERT and Transformer-XL to achieve state-of-the-art performance on 18 NLP tasks including question answering, natural language inference, sentiment analysis, and document ranking.

OpenAI's GPT is an unsupervised language model trained on a giant collection of free text corpora. GPT specifically uses multi-layer decoder only Transformer as well as does not generate embeddings for usage in downstream NLP tasks but fine-tunes the base model itself. The successor to GPT and GPT-2, GPT-3 is similar to GPT-2, but it uses alternating dense and sparse attention patterns as in Sparse Transformer [Child et al. 2019]. This large scale transformer-based language model has been trained on 175 billion parameters and has shown strong performance for over two dozen NLP tasks. GPT-3 is the largest model so far, and its impressive capabilities have positioned it to outrank other pre-trained models.

6.2 Computer Vision (CV)

Visual attention has become popular in many main stream CV tasks to focus on relevant regions within the image, and capture structural long-range dependencies between parts of the image. Visual attention term was conceived by Mnih et al. [2014] in which attention was proposed for the *Image Classification* task. Here the authors use attention to not only select relevant regions and locations within the input image but also to reduce computational complexity of CNNs by processing only selected regions at high resolution. This is crucial to control the computational complexity of proposed model irrespective of the size of the input image, compared to CNNs whose computational complexity scales linearly with increase in the size of the image (number of image pixels).

Visual attention also provides a significant benefit for *Object Detection*, as it can aid to localize and recognize objects within the image. In Ba et al. [2014] authors use attention for multiple object detection problem where the image is processed in a sequential manner ("glimpse" at a time) to learn to predict one object at a time. Hence, a sequence of labels is generated in the end for multiple objects, until there are no more objects that the model can recognize. Deep Recurrent Attentive Writer [Gregor et al. 2015] exploit attention for *Image Generation*. Although it is an encoder-decoder framework that compresses and regenerates images during training; the major difference from previous work is it generates images in step by step fashion, rather than in a single pass. This is accomplished by using attention to selectively attend to parts of the input image while regenerating specific scenes within the image in an iterative manner. Self-Attention Generative Adversarial Networks [Zhang et al. 2019a] use a self-attention mechanism into convolutional GANs by calculating the response at a position as a weighted sum of the features at all positions. This helps in capturing long range dependencies efficiently compared to convolution processes alone, as they process the information in a local neighborhood. The self-attention module is complementary to convolutions by modeling long range, multi-level dependencies across image regions efficiently.

Although CNNs have become the dominant models for vision tasks since 2012, CNNs are designed specifically for images and can be computationally demanding. For next generation of efficient, scalable and domain agnostic architectures, using *Transformers* for vision tasks has become a new research direction. First major work in this direction has been the **Vision Transformer (ViT)** by Dosovitskiy et al. [2021], which directly uses the original Transformer architecture on image patches along with positional embeddings for image classification task. It has outperformed a comparable state-of-the-art CNN with four times fewer computational resources. Another work by Touvron et al. [2021] uses novel distillation approach for Transformers to perform large-scale image classification, without pre-training on an external large dataset, making it more efficient than ViT. Similarly, **Detection Transformer (DETR)** by Carion et al. [2020] is the first Object Detection framework that relied on Transformers as the main building block. DETR uses the flattened output from CNNs and positional encodings as input to directly generate output classes and bounding boxes. It also streamlined the pipeline by removing many hand designed components (e.g., anchor generation, non-maximum suppression) required in existing approaches. Zhu et al. [2021] further use deformable attention module to reduce the training time of DETR by 10×, lowering its computational cost. Finally, Transformers have also been used for Image Generation task with Image Transformer by Parmar et al. [2018] and Image GPT by Chen et al. [2020], designed to sequentially predict each pixel of an output image given its previously generated pixels.

6.3 Multi-Modal Tasks

Attention has been extensively used for multi-modal applications, because it helps to understand relationships between different modalities. *Multimedia Description* is the task of generating a natural language text description of a multimedia input sequence that can be image and video [Cho et al. 2015]. Similarly to QA, here attention performs the function of finding relevant parts of the input image [Xu et al. 2015] to predict the next word in caption or focus on smaller subset of frames for video description [Yao et al. 2015].

For *Speech Recognition*, in Chan et al. [2016] the authors claim that without attention, the model significantly overfits the data, because it memorizes the training transcripts, without really paying attention to the acoustics. Chorowski et al. [2015] also describe how Speech Recognition differs from other NLP and CV tasks as the input is much noisier, lacks a clear structure and has multiple similar speech fragments. In this work, authors propose an attention mechanism such that it takes into account both the location and content of the important fragments in the input sequence. Adapting the attention mechanism to also incorporate location helps with longer input sequences and recognition of similar/repeated speech fragments. Latest work by Qin and Qu [2020] propose a novel approach for visually inspecting the encoder representations so that they can be used to understand how attention mechanism works for speech recognition task.

Another interesting work on *Human Communication Comprehension* [Zadeh et al. 2018] addresses the challenging problem of comprehending face-to-face communication that is a complex multi-modal task involving language, vision and speech modalities simultaneously. Here attention is specifically used for discovering interactions between different modalities (called cross-view dynamics) at each time step. The authors demonstrated that the approach shows state-of-the-art performance on multiple tasks such as speaker trait recognition and emotion recognition.

Again, Transformers have also been used extensively for vision-language tasks to learn generic representations that can effectively encode cross-modal relationships. The two main types of Transformers used in this domain are: single and multi stream [Khan et al. 2021]. In single stream models such as videoBERT [Sun et al. 2019] and visualBERT [Li et al. 2019b], all multi-model inputs are given to a single Transformer as input, that automatically discovers relationships between the two domains. However, ViLBERT [Lu et al. 2019] and LXMERT [Tan and Bansal 2019] fall in the multi-stream category, where independent Transformers for used for each modality first and later cross-modal representations are learned using another co-attentional Transformer.

6.4 Recommender Systems

Attention has seen a significant usage in recommender systems for user profiling, learning user/item representations, and exploiting auxiliary information such as knowledge graph, social network. User profiling aims to assign attention weights to interacted items of a user to capture long and short term interests in a more effective manner. This is intuitive, because all interactions of a user are not relevant for the recommendation of an item and user's interests are transient as well as varied in the long and short time span. Attention mechanism has been used for finding the most relevant items in user's history to improve recommendations in Collaborative Filtering

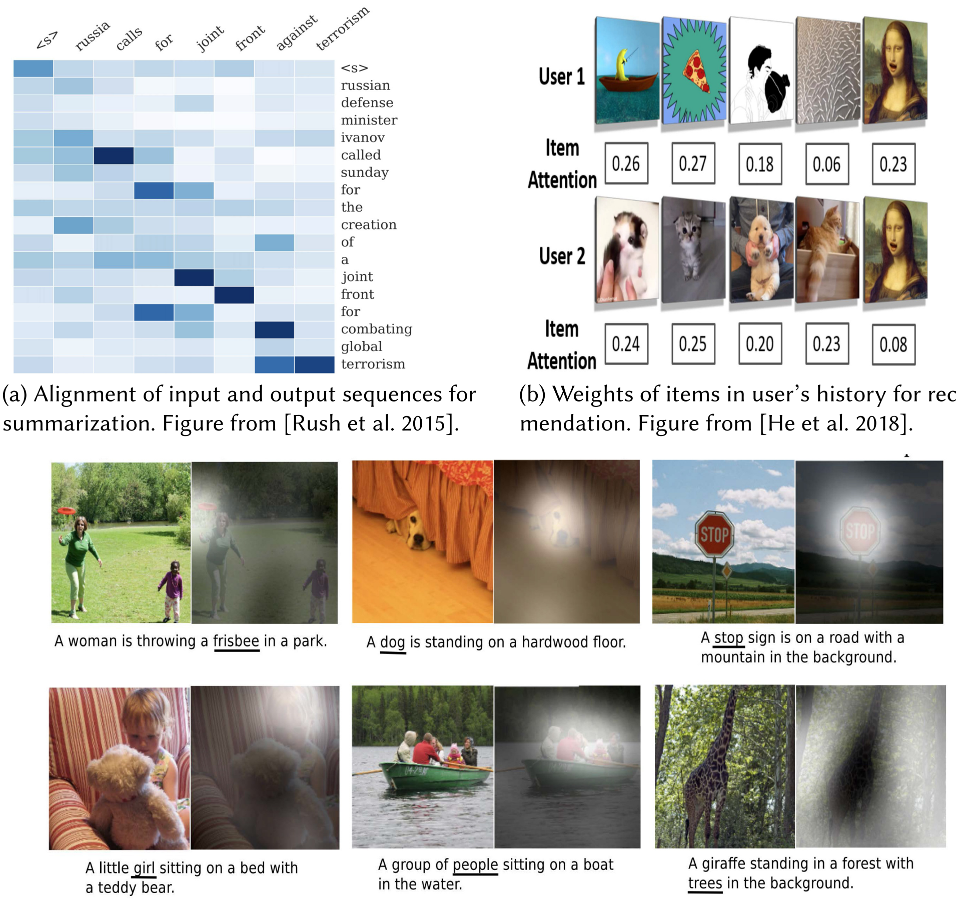
[He et al. 2018; Shuai Yu 2019] as well as RNN-based sequential models [Kang and McAuley 2018; Qiannan et al. 2019; Ying et al. 2018b; Zhou et al. 2018].

Learning effective user and item representations lies at the heart of recommender systems. Consequently, Zhu et al. [2020] use attention for cross domain recommendation problem, in which attention is used to effectively combine user/item embeddings learned from both domains, to generate a single embedding for every user/item. Similarly, Deng et al. [2020] use HAN for learning more effective item representations for paper review rating recommendation problem. Specifically, authors use hierarchical attention to leverage the hierarchical structure of the paper reviews with three levels: sentence (level one), intra-review (level two), and inter-review (level three). Similarly, Wu et al. [2019] also use three-tier HAN to to learn user and item representations from reviews as different reviews, sentences and even words have different informativeness for modeling users and items.

Attention is also helpful to utilize auxiliary information in recommender systems more effectively. For example, Wang et al. [2019a] propose Knowledge Graph Attention Network in which a hybrid graph linking user-item interactions and item attributes is constructed to exploit higher order relations. A node's embedding is computed by learning attention weights over its neighbours that can consist of users, items or attributes. Wu et al. [2020] use HAN to attend to three key aspects that affect user's preference for image recommendation: upload history, social influence, and owner admiration. Attention is used at two levels: element level while learning individual aspect representations and aspect level, since not all aspects are equally important for learning user's preferences. Finally, recent work by Xiao et al. [2021] proposes a social recommendation model that explores higher-order friends of user in a social network to help content recommendation. The proposed method allow a user's context vector to attend to friend's context vectors so that interests and knowledge aggregated from a user's social network can be utilized for recommendation.

6.5 Graph-based Systems

Many important real-world datasets come in the form of graphs or networks; examples include social networks, knowledge graphs, protein-interaction networks, and the World Wide Web. Attention has been used to highlight elements of the graph (nodes, edges, subgraphs) that are more relevant for the main task [Lee et al. 2019]. The common approach is to compute attention-guided embeddings of nodes or edges or subgraphs or their combinations. Attention architecture in graphs is efficient, since it is parallelizable across node neighbor pair, can be applied to graph nodes having different degrees, and is directly applicable to inductive learning problems, including tasks where the model has to generalize to completely unseen graphs. In contrast to Graph Convolutional Networks, attention mechanism in graphs allows for assigning different importance to nodes of a same neighborhood, enabling a leap in model capacity. Analyzing the learned attention weights may lead to benefits in interpretability. Attention has been used in several machine learning tasks in graph-structured data including (i) *Node Classification* [Cui et al. 2020; Velickovic et al. 2018], (ii) *Link Prediction* [Zhao et al. 2017], (iii) *Graph Classification* [Lee et al. 2018], and (iv) *Graph to Sequence* Generation [Beck et al. 2018; Zheng et al. 2020]. Zhang et al. [2020] uses attention for hyperedge prediction in hypergraphs. Hypergraphs are mainly used to represent higher order interactions in graph using hyperedges, i.e., edges connecting multiple nodes. Most existing methods are designed for analyzing pairwise interactions and thus are unable to effectively capture higher-order interactions in graphs. Consequently, authors propose self-attention-based GAT for hyperedge prediction that can work with both homogeneous and heterogeneous hypergraphs with variable hyperedge size.



(a) Alignment of input and output sequences for summarization. Figure from [Rush et al. 2015].

(b) Weights of items in user's history for recommendation. Figure from [He et al. 2018].

(c) Relevant image regions for image captioning. Figure from [Xu et al. 2015].

Fig. 8. Examples of visualization of attention weights.

7 ATTENTION FOR INTERPRETABILITY

There is a growing interest in the interpretability of AI models—driven by both performance as well as transparency and fairness of models.[1] However, neural networks, particularly deep learning architectures have been criticized for their lack of interpretability [Guidotti et al. 2018].

Modeling attention is particularly interesting from the perspective of interpretability, because it allows us to directly inspect the internal working of the deep learning architectures. The hypothesis is that the magnitude of attention weights correlates with how relevant a specific region of input is for the prediction of output at each position in a sequence. This can be easily accomplished by visualizing the attention weights for a set of input and output pairs. Li et al. [2016] upholds attention as one of the important ways to explain the inner workings of neural models.

As shown in Figure 8(a), Rush et al. [2015] showed that attention model is able to focus on relevant words in the input sequence while generating output for the summarization task. In the given example, the input word *combating* has a high attention weight for the output word *against* that

[1] https://fatconference.org.

demonstrates that attention model can capture word relationships for summarization. Figure 8(b) shows attention weights can help to recognize user's interests. User 1 seems to have a preference for "cartoon" videos, while user 2 prefers videos on "animals" [He et al. 2018]. Xu et al. [2015] provide extensive list of visualizations of the relevant image regions (i.e., with high attention weights) that had a significant impact on the generated text in the image captioning task (example shown in Figure 8(c)). Similarly, Bahdanau et al. [2015] visualize attention weights that clearly show automatic alignment of sentences in French and English despite the fact that subject-verb-noun locations differ from language to language. In particular, attention model shows non-monotonic alignment by correctly aligning *environnement marin* with *marine environment.*

We also summarize a few other interesting findings as follows. De-Arteaga et al. [2019] explored gender bias in occupation classification, and showed how the words getting more attention during classification task are often gendered. Yang et al. [2016] noted that the importance of words *good* and *bad* is context dependent for determining the sentiment of the review. The authors inspected the attention weight distribution of these words to find that they span from 0 to 1, which means the model captures diverse context and assign context-dependent weight to the words. Chan et al. [2016] noted that in speech recognition, attention between character output and audio signal can correctly identify start position of the first character in audio signal and attention weights are similar for words with acoustic similarities. Finally, Kiela et al. [2018] found that the multi-representational attention assigned higher weights to GloVe, FastText word embeddings out of many other representations used, particularly GloVe for low frequency words. As another interesting application of attention, Lee et al. [2017] and Liu et al. [2018] provide a tool for visualizing attention weights of deep-neural networks. The goal is to interpret and perturb the attention weights so that one can simulate what-if scenarios and observe the changes in predictions interactively.

Despite being popularly used to shed light on inner working of black-box neural networks, using attention weights for model explainability remains an area of active research. Some articles have presented a contradictory viewpoint that challenges the usage of attention weights as explanations of model behaviour/decision making process [Jain and Wallace 2019; Serrano and Smith 2019]. Based on several experiments on application of attention models for NLP tasks, Jain and Wallace [2019] argued that attention weights are often not correlated with the typical feature importance analysis. Moreover, they performed two analyses to observe the sensitivity of predictions to the change in attention weights and observed that changing attention weights with random permutations and adversarial training do not change the output predictions. Serrano and Smith [2019] applied a different analysis based on intermediate representation erasure method and showed that attention weights are at best noisy predictors of relative importance of the specific regions of input sequence, and should not be treated as justifications for model's decisions.

8 CONCLUSION

In this survey, we discussed different ways in which attention has been formulated in the literature, and attempted to provide an overview of various techniques by discussing a taxonomy of attention, key neural network architectures using attention, and application domains that have seen significant impact. We discussed how the incorporation of attention in neural networks has led to significant gains in performance, provided greater insight into neural network's inner working by facilitating interpretability, and also improved computational efficiency by eliminating sequential processing of input. We hope that this survey provides an understanding of the different directions in which research has been done on this topic, and how techniques developed in one area can be applied to other domains. We conclude this survey with some of the emerging research directions in attention modeling.

8.1 Real-time Attention

In a usual neural machine translation model, encoding and decoding of the entire sentence happens sequentially. However, some real-time applications such as live video captions or conversations between people speaking different languages demand that machine translation model starts generating a translation before it has finished reading the entire source sentence. Chiu and Raffel [2017] use monotonic chunkwise attention that adaptively split the input sequence into smaller chunks over which soft attention is computed, thus allowing online and linear-time decoding. Ma et al. [2019] enable online decoding with Transformers by using monotonic multi-head attention that alternates between encoding and decoding. With an ever-increasing demand for real-time applications, we expect online attention to be an important area for future research.

8.2 Stand-alone Attention

Introducing attention in state-of-the-art models such as CNNs in Computer Vision has shown performance gains. The question is whether attention can be a stand-alone primitive for vision models instead of serving as an augmentation on top of convolutions. Ramachandran et al. [2019] investigated stand-alone self-attention in vision models by replacing all instances of spatial convolutions with self-attention over local regions and found that these pure self-attention vision models are able to compete with state-of-art models on benchmark vision datasets. Wang et al. [2020e] enables performing attention within a larger or even global region by factorizing 2D self-attention into two 1D self-attentions.

8.3 Model Distillation

A number of industry applications such as recommender and search systems have strict latency constraints for online model serving. While pre-trained models such as BERT have shown considerable performance improvements, they have hundreds of millions of parameters, which makes them inapplicable for online serving. The field of model distillation aims to compress an existing large, complex model with a simpler model while retaining its accuracy. Wang et al. [2020d] used deep self-attention to train a small model (student) by deeply mimicking the self-attention module of the large model (teacher). Moreover, they observed that introducing a teacher assistant (a concept introduced in Mirzadeh et al. [2020]) also helps the distillation of large pre-trained Transformer models. Similarly, Touvron et al. [2020] employed a teacher-student strategy specific to transformers ensuring that the student learns from the teacher through attention.

8.4 Attention for Interpretability

Exploring the relationship between attention weights and model interpretability continues to be an active area of research. Future research can investigate attention distributions of current models and how they can be modified to offer plausible justifications of model prediction. An example of such work is by Mohankumar et al. [2020] where they observe that in LSTM-based encoders the hidden representations at different timesteps are very similar to each other; even a random permutation of the attention weights does not affect the model's predictions. They propose a diversity-driven LSTM cell that uses an orthogonalization technique to ensure that the hidden states are farther away from each other in their spatial dimensions. These modified LSTM cells generate attention weights that (i) provide a more precise importance ranking of the hidden states, (ii) are better indicative of words important for the model's predictions, and (iii) correlate better with gradient-based attribution methods.

8.5 Auto-learning Attention

The automated design of neural network architectures using **neural architecture search (NAS)** has outperformed human designs on various tasks. An open question is if we can use NAS to search the optimal architecture of high order attention module. Ma et al. [2020] is the first attempt to extend NAS to search plug-and-play attention modules beyond the backbone architecture. They define a novel concept of attention module named Higher Order Group Attention that can represent high order attentions and utilize a differentiable search method to search the optimal attention module efficiently.

8.6 Multi-instance Attention

Existing attention mechanisms attend to individual items in the memory with a fixed granularity, e.g., a word token or a pixel in an image grid. Multi-instance attention is a generalization that allows attending to structurally adjacent group of items, e.g., 2D areas in images, or subsequences in natural language sentences. One of the first techniques working with attention over multiple instances is *area attention* [Li et al. 2019a]. A simple approach is to model the key of an area as the mean vector of the key of each item in the area. Alternatively, a richer representation of each area can be formed by using derived features such as standard deviation of the key vectors within each area. Such formulations can be very useful in exploring attention mechanism on groups of items of dynamic shapes and sizes.

8.7 Multi-agent Systems

Understanding and modeling behavior of multi-agent systems is required in many real-world applications, including autonomous vehicles, multi-player games, and so on. Attention mechanism working with deep generative models can be used to model interactions within multi-agent systems. Li et al. [2020a] and Fujii et al. [2020] use attention to capture behavior generating process of multi-agent systems, as well as identify agent groups and how they interact with each other.

8.8 Scalability

Large Transformer models have shown extraordinary success in achieving state-of-the-art results in many NLP and computer vision applications. However, training and deploying these models can be prohibitively costly for long sequences (such as in bioinformatics) as the standard self-attention mechanism of the Transformer uses $O(n^2)$ time and space with respect to sequence length. An important theme of research is to reduce the quadratic time and space complexity of Transformers to linear without loss in performance of these models. Sukhbaatar et al. [2019b] propose an approach that uses a learnable masking function to dynamically control attention span in self-attention mechanism. This allows to increase the maximum context size used in Transformers, without increasing computational cost. Choromanski et al. [2021] introduce Performers, which approximate softmax full-rank attention in linear space and time complexity. Wang et al. [2020a] demonstrate that the self-attention mechanism can be approximated by a low-rank matrix and exploit this finding to propose a new self-attention mechanism, which reduces the overall self-attention complexity from quadratic to linear in both time and space. The resulting linear transformer, the Linformer, performs on par with standard Transformer models. Li et al. [2020b] propose an *Edge Predictor* that utilizes an LSTM model to dynamically predict attention edges (relationship between tokens), thus automatically searching for the best attention patterns in long sequences.

REFERENCES

Sami Abu-El-Haija, Bryan Perozzi, Rami Al-Rfou, and Alexander A. Alemi. 2018. Watch your step: Learning node embeddings via graph attention. In *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 9180–9190.

Artaches Ambartsoumian and Fred Popowich. 2018. Self-attention: A better building block for sentiment analysis neural network classifiers. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Association for Computational Linguistics, 130–139.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 6077–6086.

Lei Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. 2014. Multiple object recognition with visual attention. In *Proceedings of the International Conference on Learning Representations*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*.

Daniel Beck, Gholamreza Haffari, and Trevor Cohn. 2018. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 273–283.

Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. 2017. Massive exploration of neural machine translation architectures. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1442–1451.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, and Ilya 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, Vol. 33. 1877–1901.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, Lecture Notes in Computer Science, Vol. 12346). Springer, 213–229.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 4960–4964.

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. 2020. Generative pre-training from pixels. In *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 119. PMLR, 1691–1703.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. arXiv:1904.10509. Retrieved from https://arxiv.org/abs/1904.10509.

Chung-Cheng Chiu and Colin Raffel. 2017. Monotonic chunkwise attention. arXiv:1712.05382. Retrieved from https://arxiv.org/abs/1712.05382.

Kyunghyun Cho, Aaron Courville, and Yoshua Bengio. 2015. Describing multimedia content using attention-based encoder-decoder networks. *IEEE Trans. Multimedia* 17, 11 (2015), 1875–1886.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of 8th Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*. Association for Computational Linguistics, 103–111.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1724–1734.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 93–98.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J. Colwell, and Adrian Weller. 2021. Rethinking attention with performers. In *Proceedings of the International Conference on Learning Representations*.

Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Advances in Neural Information Processing Systems*. MIT Press, 577–585.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? An analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, 276–286.

Limeng Cui, Haeseung Seo, Maryam Tabar, Fenglong Ma, Suhang Wang, and Dongwon Lee. 2020. DETERRENT: Knowledge guided graph attention network for detecting healthcare misinformation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Association for Computing Machinery, 492–502.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2978–2988.

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. Association for Computing Machinery, 120–128.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. 2019. Universal transformers. In *Proceedings of the 7th International Conference on Learning Representations*.

Zhongfen Deng, Hao Peng, Congying Xia, Jianxin Li, Lifang He, and Philip Yu. 2020. Hierarchical bi-directional self-attention networks for paper review rating recommendation. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, 6302–6314.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 4171–4186.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations*.

Jun Feng, Minlie Huang, Yang Yang, and Xiaoyan Zhu. 2016. GAKE: Graph aware knowledge embedding. In *Proceedings of the 26th International Conference on Computational Linguistics*. The COLING 2016 Organizing Committee, 641–651.

Keisuke Fujii, Naoya Takeishi, Yoshinobu Kawahara, and Kazuya Takeda. 2020. Policy learning with partial observation and mechanical constraints for multi-person modeling. arXiv:2007.03155. Retrieved from https://arxiv.org/abs/2007.03155.

Andrea Galassi, Marco Lippi, and Paolo Torroni. 2020. Attention in natural language processing. *IEEE Trans. Neural Netw. Learn. Syst.* (2020), 1–18.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014a. Neural turing machines. arXiv:1410.5401. Retrieved from https://arxiv.org/abs/1410.5401.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014b. Neural turing machines. arXiv:1410.5401. Retrieved from https://arxiv.org/abs/1410.5401.

Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra. 2015. DRAW: A recurrent neural network for image generation. Proceedings of Machine Learning Research, Vol. 37. PMLR, 1462–1471.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *Comput. Surv.* 51, 5, Article 93 (2018), 42 pages.

Xiangnan He, Zhankui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. 2018. NAIS: Neural attentive item similarity model for recommendation. *IEEE Trans. Knowl. Data Eng.* 30, 12 (2018), 2354–2366. https://doi.org/10.1109/TKDE.2018.2831682

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 1693–1701.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*. Association for Computational Linguistics, 3543–3556.

Saumya Jetley, Nicholas A. Lord, Namhoon Lee, and Philip Torr. 2018. Learn to pay attention. In *Proceedings of the International Conference on Learning Representations.*

Wang-Cheng Kang and Julian J. McAuley. 2018. Self-attentive sequential recommendation. In *Proceedings of the IEEE International Conference on Data Mining*. IEEE Computer Society, 197–206.

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. 2021. Transformers in vision: A survey. arXiv:2101.01169. Retrieved from https://arxiv.org/abs/2101.01169.

Douwe Kiela, Changhan Wang, and Kyunghyun Cho. 2018. Dynamic meta-embeddings for improved sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1466–1477.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *Proceedings of the 8th International Conference on Learning Representations.*

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*. Association for Computational Linguistics, 7057–7075.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *Proceedings of the 33rd International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 48. PMLR, 1378–1387.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of the International Conference on Learning Representations*.

Jaesong Lee, Joong-Hwi Shin, and Jun-Seok Kim. 2017. Interactive visualization and manipulation of attention-based neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, 121–126.

John Boaz Lee, Ryan Rossi, and Xiangnan Kong. 2018. Graph classification using structural attention. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 1666–1674.

John Boaz Lee, Ryan A. Rossi, Sungchul Kim, Nesreen K. Ahmed, and Eunyee Koh. 2019. Attention models in graphs: A survey. *ACM Trans. Knowl. Discov. Data* 13, 6, Article 62 (2019), 25 pages. https://doi.org/10.1145/3363574

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 7871–7880.

Guangyu Li, Bo Jiang, Hao Zhu, Zhengping Che, and Yan Liu. 2020a. Generative attention networks for multi-agent behavioral modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 7195–7202.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *CoRR* abs/1612.08220.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019b. VisualBERT: A simple and performant baseline for vision and language. arXiv:1908.03557. Retrieved from https://arxiv.org/abs/1908.03557.

Xiaoya Li, Yuxian Meng, Mingxin Zhou, Qinghong Han, Fei Wu, and Jiwei Li. 2020b. Sac: Accelerating and structuring self-attention via sparse adaptive connection. arXiv:2003.09833. Retrieved from https://arxiv.org/abs/2003.09833.

Yang Li, Lukasz Kaiser, Samy Bengio, and Si Si. 2019a. Area attention. In *Proceedings of the International Conference on Machine Learning*. PMLR, 3846–3855.

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *Proceedings of the International Conference on Learning Representations* (2017).

Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. 2020. SPACE: Unsupervised object-oriented scene representation via spatial attention and decomposition. In *Proceedings of the 8th International Conference on Learning Representations*.

Shusen Liu, Tao Li, Zhimin Li, Vivek Srikumar, Valerio Pascucci, and Peer-Timo Bremer. 2018. Visual interrogation of attention-based models for natural language inference and machine comprehension. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, 36–41.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. arXiv:1907.11692. Retrieved from https://arxiv.org/abs/1907.11692.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, Vol. 32.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., 289–297.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015a. Effective approaches to attention-based neural machine translation. arXiv:1508.04025. Retrieved from https://arxiv.org/abs/1508.04025.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015b. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1412–1421.

Benteng Ma, Jing Zhang, Yong Xia, and Dacheng Tao. 2020. Auto learning attention. *Adv. Neural Inf. Process. Syst.* 33 (2020).

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017b. Interactive attention networks for aspect-level sentiment classification. arXiv:1709.00893. Retrieved from https://arxiv.org/abs/1709.00893.

Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. 2017a. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 1903–1911.

Xutai Ma, Juan Pino, James Cross, Liezl Puzon, and Jiatao Gu. 2019. Monotonic multihead attention. arXiv:1909.12406. Retrieved from https://arxiv.org/abs/1909.12406.

Yukun Ma, Haiyun Peng, and Erik Cambria. 2018. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. AAAI Press, 5876–5883.

Suraj Maharjan, Manuel Montes, Fabio A. González, and Thamar Solorio. 2018. A genre-aware attention model to improve the likability prediction of books. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 3381–3391.

Andre Martins and Ramon Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proceedings of the International Conference on Machine Learning*. PMLR, 1614–1623.

André F. T. Martins, Marcos Treviso, António Farinhas, Vlad Niculae, Mário A. T. Figueiredo, and Pedro M. Q. Aguiar. 2020. Sparse and continuous attention mechanisms. arXiv:2006.07214. Retrieved from https://arxiv.org/abs/2006.07214.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, Vol. 32.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. 2020. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 5191–5198.

Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. 2014. Recurrent models of visual attention. In *Proceedings of the 27th International Conference on Neural Information Processing Systems, Volume 2*. MIT Press, 2204–2212.

Akash Kumar Mohankumar, Preksha Nema, Sharan Narasimhan, Mitesh M. Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. 2020. Towards transparent and explainable attention models. arXiv:2004.14243. Retrieved from https://arxiv.org/abs/2004.14243.

Elizbar A. Nadaraya. 1964. On estimating regression. *Theory Probab. Appl.* 9, 1 (1964), 141–142.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 280–290.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. 2018. Image transformer. Proceedings of Machine Learning Research, Vol. 80. 4055–4064.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017. Deeper attention to abusive user content moderation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 1125–1135.

Ben Peters, Vlad Niculae, and André FT Martins. 2019. Sparse sequence-to-sequence models. arXiv:1905.05702. Retrieved from https://arxiv.org/abs/1905.05702.

Zhu Qiannan, Xiaofei Zhou, Zeliang Song, Jianlong Tan, and Li Guo. 2019. DAN: Deep attention neural network for news recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 5973–5980.

C. Qin and D. Qu. 2020. Towards understanding attention-based speech recognition models. *IEEE Access* 8 (2020), 24358–24369.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language models are unsupervised multitask learners. https://openai.com/blog/better-language-models/.

Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. 2019. Stand-alone self-attention in vision models. arXiv:1906.05909. Retrieved from https://arxiv.org/abs/1906.05909.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 379–389.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable?. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2931–2951.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2018. Disan: Directional self-attention network for rnn/cnn-free language understanding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*.

Min Yang, Baocheng Li, Qiang Qu, Jialie Shen, Shuai Yu, and Yongbo Wang. 2019. NAIRS: A neural attentive interpretable recommendation system. *The Web Conference* (2019).

Alessandro Sordoni, Philip Bachman, Adam Trischler, and Yoshua Bengio. 2016. Iterative alternating neural attention for machine reading. arXiv:1606.02245. Retrieved from https://arxiv.org/abs/1606.02245.

Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. 2019a. Adaptive attention span in transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 331–335.

Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. 2019b. Adaptive attention span in transformers. arXiv:1905.07799. Retrieved from https://arxiv.org/abs/1905.07799.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 2440–2448.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019. VideoBERT: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.*

Qiang Sun and Yanwei Fu. 2019. Stacked self-attention networks for visual question answering. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*. Association for Computing Machinery, 207–211.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. Association for Computational Linguistics, 5099–5110.

Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 214–224.

Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. 2018. Why self-attention? A targeted evaluation of neural machine translation architectures. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 4263–4272.

Yi Tay, Anh Tuan Luu, Aston Zhang, Shuohang Wang, and Siu Cheung Hui. 2019. Compositional de-attention networks. *Adv. Neural Inf. Process. Syst.* 32 (2019), 6135–6145.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2020. Training data-efficient image transformers & distillation through attention. arXiv:2012.12877. Retrieved from https://arxiv.org/abs/2012.12877.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2021. Training data-efficient image transformers & distillation through attention. arXiv:2012.12877. Retrieved from https://arxiv.org/abs/2012.12877.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc., 5998–6008.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations.*

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, Vol. 28. MIT Press, 2692–2700.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 5797–5808.

Feng Wang and David MJ Tax. 2016. Survey on the attention based RNN model and its applications in computer vision. arXiv:1601.06823. Retrieved from https://arxiv.org/abs/1601.06823.

Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. 2020e. Axial-deeplab: Standalone axial-attention for panoptic segmentation. In *Proceedings of the European Conference on Computer Vision*. Springer, 108–126.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020c. Relational graph attention network for aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, 3229–3238.

Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and H. Linformer Ma. 2020a. Self-attention with linear complexity. arXiv:2006.04768. Retrieved from https://arxiv.org/abs/2006.04768.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2017. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. AAAI Press, 3316–3322.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020d. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. arXiv:2002.10957. Retrieved from https://arxiv.org/abs/2002.10957.

Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019a. KGAT: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Association for Computing Machinery, 950–958.

Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. 2019b. Heterogeneous graph attention network. In *Proceedings of the World Wide Web Conference*. 2022–2032.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 606–615.

Yue Wang, Jing Li, Michael Lyu, and Irwin King. 2020b. Cross-media keyphrase prediction: A unified framework with multi-modality multi-head attention and image wordings. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 3311–3324.

Geoffrey S. Watson. 1964. Smooth regression analysis. *Ind. J. Stat. Ser. A* (1964), 359–372.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. arXiv:1410.3916. Retrieved from https://arxiv.org/abs/1410.3916.

Chuhan Wu, Fangzhao Wu, Junxin Liu, and Yongfeng Huang. 2019. Hierarchical user and item representation with three-tier attention for recommendation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 1818–1826.

Le Wu, Lei Chen, Richang Hong, Yanjie Fu, Xing Xie, and Meng Wang. 2020. A hierarchical attention model for social contextual image recommendation. *IEEE Trans. Knowl. Data Eng.* 32, 10 (2020), 1854–1867.

Wenyi Xiao, Huan Zhao, Haojie Pan, Yangqiu Song, Vincent W. Zheng, and Qiang Yang. 2021. Social explorative attention based recommendation for content distribution platforms. *Data Min. Knowl. Discov.* 35, 2 (2021), 533–567.

Caiming Xiong, Stephen Merity, and Richard Socher. 2016. Dynamic memory networks for visual and textual question answering. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning, Volume 48*. JMLR.org, 2397–2406.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. *Proceedings of Machine Learning Research,* Vol. 37. 2048–2057.

Song Xu, Haoran Li, Peng Yuan, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Self-attention guided copy mechanism for abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 1355–1362.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R. Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, Vol. 32.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1480–1489.

Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. 2015. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE Computer Society, 4507–4515.

Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018a. Sequential recommender system based on hierarchical attention network. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press, 3926–3932.

Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018b. Sequential recommender system based on hierarchical attention networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3926–3932.

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2018. Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag.* 13, 3 (2018), 55–75.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6281–6290.

Amir Zadeh, Paul Pu Liang, Soujanya Poria, Prateek Vij, Erik Cambria, and Louis-Philippe Morency. 2018. Multi-attention recurrent network for human communication comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 5642–5649.

B. Zhang, D. Xiong, J. Xie, and J. Su. 2020. Neural machine translation with GRU-gated attention model. *IEEE Trans. Neural Netw. Learn. Syst.* 31, 11 (2020), 4688–4698.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. 2019a. Self-attention generative adversarial networks. In *Proceedings of the International Conference on Machine Learning*. 7354–7363.

Ruochi Zhang, Yuesong Zou, and Jian Ma. 2020. Hyper-SAGNN: A self-attention based graph neural network for hypergraphs. In *Proceedings of the International Conference on Learning Representations*.

Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019b. Deep learning based recommender system: A survey and new perspectives. *Comput. Surv.* 52, 1, Article 5 (2019).

Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. 2020. Exploring self-attention for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Shenjian Zhao and Zhihua Zhang. 2018. Attention-via-attention neural machine translation. In *Association for the Advancement of Artificial Intelligence*.

Zhou Zhao, Ben Gao, Vicent W. Zheng, Deng Cai, Xiaofei He, and Yueting Zhuang. 2017. Link prediction via ranking metric dual-level attention network learning. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 3525–3531.

Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. GMAN: A graph multi-attention network for traffic prediction. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*. AAAI Press, 1234–1241.

Chang Zhou, Jinze Bai, Junshuai Song, Xiaofei Liu, Zhengchao Zhao, Xiusi Chen, and Jun Gao. 2018. ATRank: An attention-based user behavior modeling framework for recommendation. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. AAAI Press, 4564–4571.

Feng Zhu, Yan Wang, Chaochao Chen, Guanfeng Liu, and Xiaolin Zheng. 2020. A graphical and attentional framework for dual-target cross-domain recommendation. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*. 3001–3008.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. 2021. Deformable DETR: Deformable transformers for end-to-end object detection. In *Proceedings of the International Conference on Learning Representations*.

Received November 2020; revised April 2021; accepted May 2021