

An Attentive Survey of Attention Models

SNEHA CHAUDHARI, LinkedIn Corporation, USA

VARUN MITHAL, LinkedIn Corporation, USA

GUNGOR POLATKAN, LinkedIn Corporation, USA

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 which 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. Finally, we also describe how attention has been used to improve the interpretability of neural networks. 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. 2020. An Attentive Survey of Attention Models. *J. ACM* 37, 4, Article 111 (December 2020), 20 pages. <https://doi.org/10.1145/1122445.1122456>

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 [Galassi et al. 2020], Speech [Cho et al. 2015] and Computer Vision [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 translation and summarization tasks, only certain words in the input sequence may be relevant for predicting the next word. Likewise, in an image captioning problem, some regions of the input image may be more relevant for generating the next word in the caption. AM incorporates this notion of relevance by allowing the model to dynamically *pay attention to*

Authors' addresses: Sneha Chaudhari, snchaudhari@linkedin.com, LinkedIn Corporation, 700 E Middlefield Rd, Mountain View, California, USA, 94043; Varun Mithal, LinkedIn Corporation, 700 E Middlefield Rd, Mountain View, California, USA, 94043, vamithal@linkedin.com; Gungor Polatkan, LinkedIn Corporation, 700 E Middlefield Rd, Mountain View, California, USA, 94043; Rohan Ramanath, LinkedIn Corporation, 700 E Middlefield Rd, Mountain View, California, USA, 94043.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Association for Computing Machinery.

0004-5411/2020/12-ART111 \$15.00

<https://doi.org/10.1145/1122445.1122456>

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 [Young et al. 2018] for multiple tasks such as Machine Translation, Question Answering, Sentiment Analysis, and Part-of-Speech tagging. 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).

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].

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] 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 paper in Section 8.

Related surveys: There have been a few domain-specific surveys on attention focusing on Computer Vision [Wang and Tax 2016], and 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 & 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 Nadaraya-Watson in 1964 [Nadaraya 1964; Watson 1964]. We are given a training data of n instances comprising features and their corresponding target values $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$. We want to predict the target value \hat{y} for a new query instance x . A naive estimator will predict the simple average of

target values of all training instances: $\hat{y} = \frac{1}{n} \sum_{i=1}^n y_i$. Nadaraya-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 $\{x_1, x_2, \dots, x_T\}$, where T is the length of input sequence, and encodes it into fixed length vectors $\{h_1, h_2, \dots, h_T\}$. The decoder is also an RNN which then takes a single fixed length vector h_T as its input and generates an output sequence $\{y_1, y_2, \dots, y_{T'}\}$ 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 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 $\{h_1, h_2, \dots, h_T\}$. 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, which we refer to as candidate state) and s_{j-1} (the decoder hidden state, which we refer to as query 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,

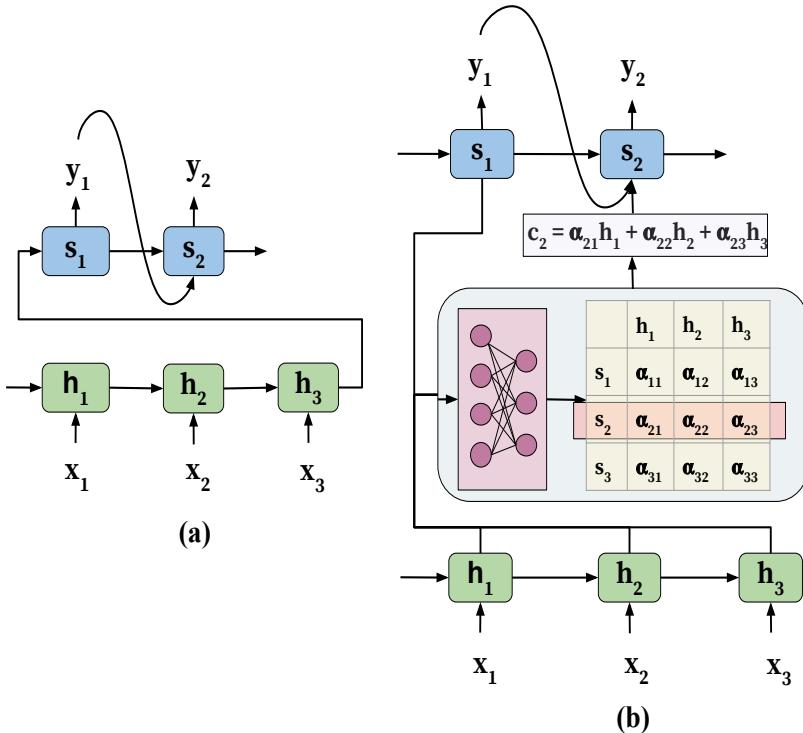


Fig. 2. Encoder-decoder architecture: (a) traditional (b) with attention model

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	$y_j = g(y_{j-1}, s_j, c)$	$y_j = g(y_{j-1}, s_j, c_j)$

$x = (x_1, \dots, x_T)$: 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

Table 1. Encoder-decoder architecture: traditional and with attention model

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 (candidate state) and s_{j-1} (query state) which 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 candidate state h_i for the query state s_{j-1} . This alignment function outputs energy scores e_{ij} which are then fed into the distribution function (denoted by p in Table 1) which converts the energy scores into attention weights. This distribution function most generally is the softmax function, but we refer the reader to [Galassi et al. 2020] for a discussion of several types of functions the can be used to compute the attention weights. When the functions a and p are differentiable, the whole attention based encoder-decoder 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 which 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:

$$A(q, K, V) = \sum_i p(a(k_i, q)) * v_i \quad (1)$$

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.

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. These 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. 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 2. 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 3.

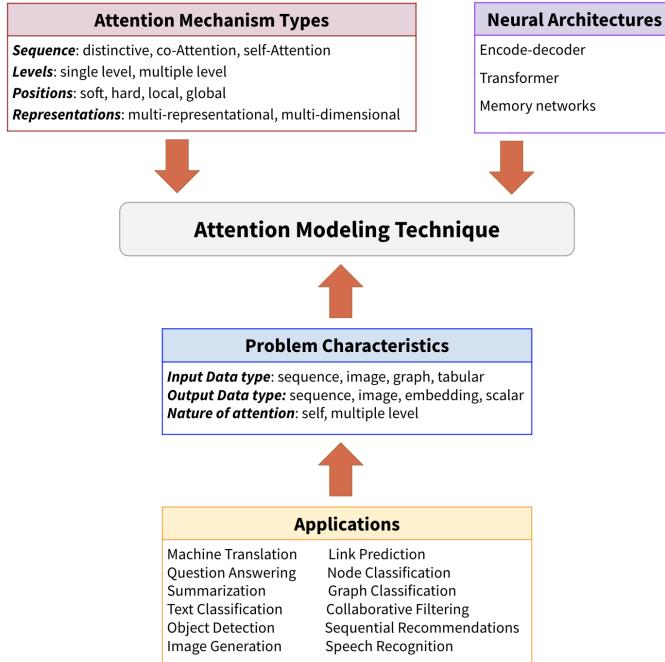


Fig. 3. Key components of Attention Modeling Techniques

4.1 Number of sequences

Thus far we have only considered the case which involves a single input and corresponding output sequence. This type of attention, which we refer to as **distinctive**, is used when candidate 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 essentially helps to simultaneously detect key phrases in the question and corresponding regions of images relevant to the answer.

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 query and candidate 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, 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.

Category	Type
Number of Sequences	distinctive, co-attention, self
Number of Abstraction Levels	single-level, multi-level
Number of Positions	soft/global, hard, local
Number of Representations	multi-representational, multi-dimensional

Table 2. Categories and types of attention within each category.

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. 2015]	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. 2018]	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

Table 3. Summary of key papers for technical approaches in AMs. ‘-’ means not applicable.

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**. On the other hand, 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 which 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 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 locate the exact regions in the image which 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.

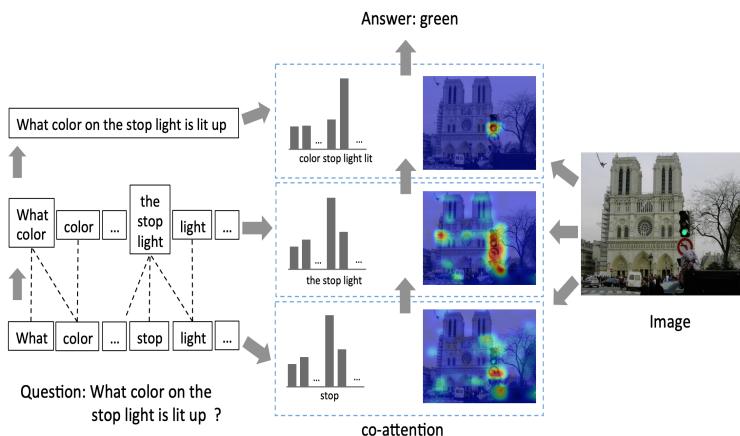


Fig. 4. The AM proposed by [Lu et al. 2016] for Visual Question Answering task which is a combination of co-attention (visual and text) and multi-level (word level, phrase level and question level) attention.

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 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. 2015] 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, on the other hand, 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 trade-off 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 three salient neural architectures used in conjunction with attention: (1) the Encoder-Decoder framework, (2) the Transformer which circumvents the sequential processing component of recurrent models with the use of attention, and (3) Memory Networks which extend attention beyond a single input sequence. 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 apriori (eg. in case of natural language modeling). The authors achieved this using attention weights to model the probability of choosing the i^{th} 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].

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.

Transformer architecture has become state-of-the-art approach for many mainstream NLP 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. Within the last 3 years, multiple variants of Transformer have been adopted for a wide variety of problems such as OpenAI's GPT and GPT-2 (decoder-only transformers for Language Modeling) [Radford et al. 2018], Bidirectional Encoder Representations from Transformer(BERT) for Language Representations [Devlin et al. 2019], Image Transformer for Image Generation [Parmar et al. 2018],

Universal Transformer for Question Answering [Dehghani et al. 2019] and Reinforcement Learning [Parisotto et al. 2019] to name a few.

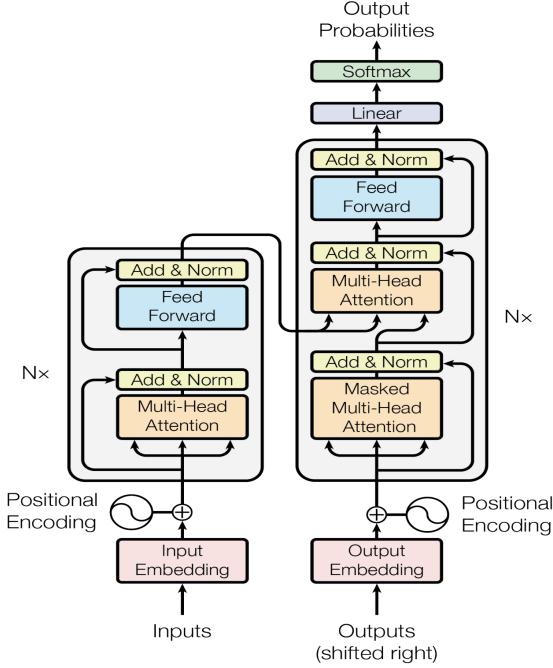


Fig. 5. Transformer Architecture. Figure from [Vaswani et al. 2017].

The Transformer architecture is shown in Figure 5. It is composed of a stack of 6 identical layers of encoders and decoders with two sub-layers: Feed Forward Network(FFN) layer and multi-head self attention layer. The self attention is used within each sub-layer to relate tokens and their positions within the same input sequence. Further, attention is known as *multi-head*, because several attention layers are stacked in parallel, with different linear transformations of the same input. This helps the model to capture various aspects of the input and improves its expressiveness. Feed Forward Network is applied independently to each position in the input sequence, increasing parallel processing. The positional encoding is used because input is sequential which demands the model to make use of the temporal aspect of the input, 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.

5.3 Memory Networks

Applications like Question Answering 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. End-to-End Memory Networks [Sukhbaatar et al. 2015] achieve this by using an array of memory blocks to store the database of facts, and using attention to model relevance of each fact in the memory for answering the given query. Using attention also provides computational advantage by making the objective continuous and enabling

end-to-end training via backpropagation. End-to-End Memory Networks can be considered as a generalization of AM, wherein instead of modeling attention over a single sequence they model it over a large database of sequences (facts). Hence, we can think of memory networks as 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).

Neural Turing Machine (NTM) [Graves et al. 2014] 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. Unlike Memory networks, the NTM memory uses both content and address-based access. This allows NTM to be trained end-to-end to infer simple algorithms such as copying, sorting and associative recall from input and output examples.

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, document representations/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 4 which can become a starting point for further investigation. We describe attention modeling in the following application domains: (i) Natural Language Processing(NLP), (ii) Computer Vision, (iii) Multi-Modal Tasks, (iv) Graphical Systems and (v) Recommender Systems.

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 which 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.

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

Application Domain	Application	Seminal Works
Natural Language Processing	Machine Translation	[Bahdanau et al. 2015], [Luong et al. 2015], [Vaswani et al. 2017], [Britz et al. 2017], [Tang et al. 2018]
	Question Answering	[Hermann et al. 2015], [Sukhbaatar et al. 2015]
	Summarization	[Rush et al. 2015], [Nallapati et al. 2016], [Chopra et al. 2016]
	Text Classification	[Yang et al. 2016]
	Text Representation Learning	[Devlin et al. 2019], [Radford et al. 2018], [Lin et al. 2017], [Kiela et al. 2018]
	Sentiment Analysis	[Wang et al. 2016], [Ma et al. 2018], [Tang et al. 2016], [Ambartsoumian and Popowich 2018]
Computer Vision	Image Classification	[Mnih et al. 2014], [Jetley et al. 2018]
	Image Generation	[Gregor et al. 2015], [Parmar et al. 2018]
	Object Detection	[Ba et al. 2014]
	Image Synthesis	[Zhang et al. 2019]
Multi-Modal Tasks	Multimedia (Image, Video) Description	[Xu et al. 2015], [Yao et al. 2015], [Cho et al. 2015]
	Speech Recognition	[Chorowski et al. 2015], [Chan et al. 2016]
	Visual Question Answering	[Lu et al. 2016], [Anderson et al. 2018]
	Human Communication Comprehension	[Zadeh et al. 2018]
Graph-based Systems	Graph Classification	[Lee et al. 2018], [Ma et al. 2017]
	Graph to Sequence Generation	[Beck et al. 2018]
	Node Classification	[Velickovic et al. 2018], [Abu-El-Haija et al. 2018], [Feng et al. 2016]
	Link Prediction	[Zhao et al. 2017]
Recommender System	Collaborative Filtering	[He et al. 2018], [Shuai Yu 2019]
	Sequential Recommendations	[Zhou et al. 2018], [Kang and McAuley 2018]

Table 4. Summary of key applications of AMs

to be challenging in the past for the task of summarization but the proposed method showed significant gains compared to several existing baselines.

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

have also looked at some applications of Transformer in NLP domain for language modeling and representations such as OpenAI's GPT [Radford et al. 2018], BERT [Devlin et al. 2019], and Transformer-XL [Dai et al. 2019].

Similarly, 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. 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].

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 used 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 used 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 (DRAW) [Gregor et al. 2015] exploited attention for **Image Generation**. Although it is an encoder-decoder framework which 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.

Lastly, Self-Attention Generative Adversarial Networks (SAGANs) [Zhang et al. 2019] 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.

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 which can be image and video [Cho et al. 2015]. Similar 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 proposed 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.

Another interesting work on **Human Communication Comprehension** [Zadeh et al. 2018] addresses the challenging problem of comprehending face-to-face communication which 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.

6.4 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) which 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** [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].

6.5 Recommender Systems

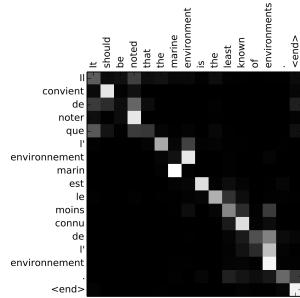
Attention has also been used in recommender systems for user profiling, i.e., assigning attention weights to interacted items of a user to capture long and short term interests in a more effective manner. This is intuitive because all interactions of a user are not relevant for the recommendation of an item and user's interests are transient as well as varied in the long and short time span. Self attention mechanism has been used for finding the most relevant items in user's history to improve item recommendations either with **Collaborative Filtering** framework [He et al. 2018; Shuai Yu 2019], or within an encoder-decoder architecture for **Sequential Recommendations** [Kang and McAuley 2018; Zhou et al. 2018].

7 ATTENTION FOR INTERPRETABILITY

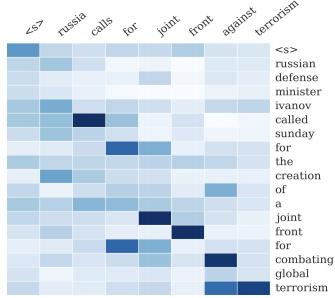
There is a growing interest in the interpretability of AI models - driven by both performance as well as transparency and fairness of models¹. However, neural networks, particularly deep learning architectures have been criticized for their lack of interpretability [Guidotti et al. 2018].

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

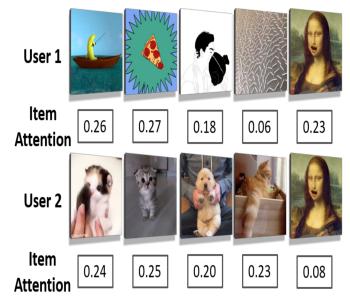
¹<https://fatconference.org>



(a) Alignment of French and English sentences in MT [Bahdanau et al. 2015]



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



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



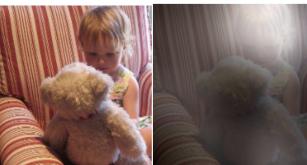
A woman is throwing a frisbee in a park.



A dog is standing on a hardwood floor.



A stop sign is on a road with mountain in the background.



A little girl sitting on a bed with a teddy bear.



A group of people sitting on a boat in the water.



A giraffe standing in a forest with trees in the background.

(d) Relevant image regions for image captioning [Xu et al. 2015]

Fig. 6. Examples of visualization of attention weights.

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 6(a), [Bahdanau et al. 2015] visualize attention weights which 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*. Figure 6(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]. Similarly, as shown in Figure 6(c), [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* which demonstrates that attention model can capture word relationships for summarization. Finally, [Xu et al. 2015] provide extensive list of visualizations of the relevant image regions (i.e. with high attention weights) which had a significant impact on the generated text in the image captioning task (example shown in Figure 6(d)).

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 will provide a better 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.

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 Ambartsumian 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 *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. *International Conference on Learning Representations* (2014).

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *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 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1442–1451.
- William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 4960–4964.
- Kyunghyun Cho, Aaron Courville, and Yoshua Bengio. 2015. Describing multimedia content using attention-based encoder-decoder networks. *IEEE Transactions on Multimedia* 17, 11 (2015), 1875–1886.
- Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. 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.
- 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.
- 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 *7th International Conference on Learning Representations*.
- 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.
- 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.
- Andrea Galassi, Marco Lippi, and Paolo Torroni. 2020. Attention in Natural Language Processing. *IEEE Transactions on Neural Networks and Learning Systems* (2020), 1–18.
- Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401* (2014).
- 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. Surveys* 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 Transactions on Knowledge and Data Engineering* 30 (2018), 2354–2366.
- Karl Moritz Hermann, Tomas Kociský, 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 *International Conference on Learning Representations*.

- Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *IEEE International Conference on Data Mining*. IEEE Computer Society, 197–206.
- 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.
- Jaesong Lee, Joong-Hwi Shin, and Jun-Seok Kim. 2017. Interactive Visualization and Manipulation of Attention-based Neural Machine Translation. In *Proceedings of the 2017 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 Transactions on Knowledge Discovery from Data* 13, 6, Article 62 (2019), 25 pages.
- Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding Neural Networks through Representation Erasure. *CoRR* abs/1612.08220 (2016).
- Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A Structured Self-attentive Sentence Embedding. *CoRR* (2017).
- 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.
- 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.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. 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.
- Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. 2017. 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.
- 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 Thirty-Second 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 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 3381–3391.
- 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.
- Elizbar A Nadaraya. 1964. On estimating regression. *Theory of Probability and Its Applications* 9, 1 (1964), 141–142.
- Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gülcühre, 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.
- Emilio Parisotto, H. Francis Song, Jack W. Rae, Razvan Pascanu, Çağlar Gülcühre, Siddhant M. Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, Matthew M. Botvinick, Nicolas Heess, and Raia Hadsell. 2019. Stabilizing Transformers for Reinforcement Learning. (2019). arXiv:1910.06764 <http://arxiv.org/abs/1910.06764>
- Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. 2018. Image Transformer (*Proceedings of Machine Learning Research, Vol. 80*). PMLR, 4055–4064.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language Models are Unsupervised Multitask Learners. (2018).
- Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of the 2015 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 Jiale Shen Shuai Yu, Yongbo Wang. 2019. NAIRS: A Neural Attentive Interpretable Recommendation System. *The Web Conference* (2019).
- 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.
- 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.
- Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect Level Sentiment Classification with Deep Memory Network. In *Proceedings of the 2016 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 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 4263–4272.
- 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 30*. Curran Associates, Inc., 5998–6008.
- Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations*.
- Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer Networks. In *Advances in Neural Information Processing Systems 28*. MIT Press, 2692–2700.
- Feng Wang and David MJ Tax. 2016. Survey on the attention based RNN model and its applications in computer vision. *arXiv preprint arXiv:1601.06823* (2016).
- 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 Thirty-First AAAI Conference on Artificial Intelligence*. AAAI Press, 3316–3322.
- Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for Aspect-level Sentiment Classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 606–615.
- Geoffrey S Watson. 1964. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A* (1964), 359–372.
- Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention (*Proceedings of Machine Learning Research, Vol. 37*). 2048–2057.
- Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 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 2015 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. 2018. Sequential Recommender System Based on Hierarchical Attention Network. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press, 3926–3932.
- Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2018. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence magazine* 13, 3 (2018), 55–75.
- 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.
- Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. 2019. Self-attention generative adversarial networks. In *International Conference on Machine Learning*. PMLR, 7354–7363.
- 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 Yuetong 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.
- 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 Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI Press, 4564–4571.