UNICAMP
Institute of Computing

Attention as a leverage for Deep Learning

Erik de Godoy Perillo
Supervisor: Profa. Dra. Esther Luna Colombini

University of Campinas

May 27, 2019

Abstract

Attention is fundamental for intelligent beings. It is necessary for filtering the significant volumes of stimuli we constantly receive and for applying the adequate mental resources to perform tasks. Deep Learning is currently broadly applied to Artificial Intelligence. The use of Attention in Deep Learning has been increasingly frequent, resulting many times in better results. In this context, this work proposes the study and elaboration of approaches to use Attention in Deep Learning for more power and efficiency to solve problems in Artificial Intelligence. We aim at obtaining a framework generically applicable in broad problem classes such as Computer Vision, Natural Language Processing, Program Composition and others.

Contents

1 Introduction

We continually receive high volumes of multimodal stimuli from both external sources – such as visual, auditive signals – and internal sources – such as proprioception and memories. It would be very inefficient or even impossible to process all the information with the same intensity once a significant portion of it is irrelevant for the task executed at the moment and considering that we have limited actuation capacity. When we read, our vision does not focus on all words equally, but instead on a small subset of the text at a time. When we are addressing a given subject (in a train of thought), it tends to mediate the focus in the memory search process, mostly retrieving memories that are useful, whereas many other irrelevant memories are not used. It often happens that something conspicuous – such as a bird abruptly appearing in front of us or a sudden sound – quickly draws our focus, stealing it from what was previously being focused. The abilities to filter and select stimuli that are relevant for a task, to keep the focus for an extended period and to adequately direct mental processes is fundamental to human beings and other sophisticated forms of life. We name this set of abilities **Attention** [7].

Attention can potentially play an essential role in Artificial Intelligence (AI). The pursue of intelligent machines is an old effort in Computer Science [34] and is still very relevant today due to the potential to radically benefit society. Although there have been significant advancements in the field of AI, it is broadly accepted that machines still cannot perform specific complex tasks nearly as efficiently as humans or some animals and the path to achieving more intelligence is still unclear, with many different proposals [23]. Part of the problem comes from the difficulty to accurately define intelligence itself, but surveys of the works on the subject [22] suggest that a reasonably accepted concept is *the ability to perform elaborate tasks in complex and dynamic environments to achieve a wide variety of goals.* From the narrow to the broader aspects of intelligence, the functionalities of Attention are of great importance – and it increases as the level of intelligence considered increases [17].

A considerable amount of advancements in AI in recent years comes from the popularization of Deep Learning (DL) [21]. As we will discuss in the following sections, the technique mostly consists of artificial neural networks architectured in a hierarchical manner. DL showed to be effective in a variety of tasks in Computer Vision [19][16], audio processing [28] and Natural Language Processing (NLP) [36], mainly due to its ability to learn what features should be extracted (rather than relying on hand-crafted features). Along with the transposition from classic models to DL approaches, an increasingly high number of works on the field have been using concepts related to Attention in combination with DL to achieve better results. One example is image captioning (Figure 1) where the task consists of giving a natural language description of a given image. The work presented in [4] shows that the task benefits from sequentially focusing on different parts of the image in a sequence, through the use of an attentional component in the model. Other examples – which will be discussed in-depth in following sections – include linguistic translation [1], audio recognition [3] and neural computation [15]. These are evidence that concepts of Attention have indeed been useful for the field.

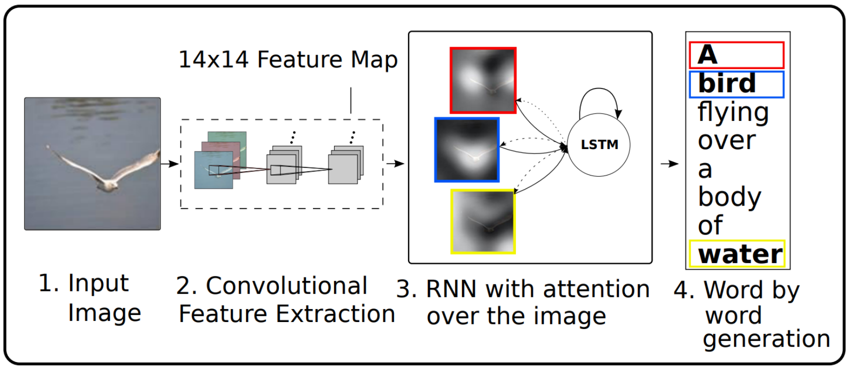

Figure 1: Diagram of natural language image description using Attention (from [4]).

1.1 Motivation and Objectives

In spite of the recent adoption of Attention by a variety of Deep Learning models and the significant improvements it has shown, we conjecture that there are still many other tasks that are still not explored. Current works also tend to focus more on the filtering functionality of Attention, but there are other aspects – such as the allocation of mental resources over time – that can be of potential benefit (we further discuss the taxonomy of Attention in following sections). Furthermore, we note that Attention models currently being used are very specific to each problem in question. Some works propose a higher level of generalization [24], but we believe it is possible to go further. Therefore, the specific objectives of this work are:

- To perform an extensive literature review on the use of Attention in modern Deep Learning;
- To identify theoretical aspects of Attention itself from areas such as psychology and neuroscience;
- To establish general elements of Attention to be applied to Deep Learning;
- To identify specific problems in different classes (robotics, vision, natural language, program composition) with improvement potential through the use of Attention;
- To propose and implement a solution based on the findings of the work to validate the ideas and evaluate them in an application.

The main contribution of the work proposed is related to the first four items: we wish to *establish a theoretical framework of Attention as a series of components and its applicabilities to Deep Learning*. Recent works show that the effort on establishing more general concepts and frameworks for Deep Learning design has been broadly useful. Examples include the ideas of *Curriculum Learning* [2] and *Generative Adversarial Networks* [12].

2 Background

2.1 Attention

The interest in the concept of Attention exists since a long time ago. Throughout the years, Attention has been studied from various perspectives [7] such as philosophy, psychology, and neurology. There are multiple definitions of the concept. In the next items, we discuss some particular aspects related to Attention.

2.1.1 A definition

We can define Attention as *the act of applying mental resources to selected stimuli following an allocation policy specific to a particular goal.* This rather broad definition captures the main concepts related to Attention: in a world with virtually infinite *stimuli* to select from the environment, agents with otherwise *finite processing resources* (but with a variety of options of *mental processes* to perform) must choose what their actions will be (and in which stimuli) in a *correct sequential manner* and in *sensible time*. As mentioned before, there is no commom definition of Attention and other works may define it differently. However, as we intend to work in a computational perspective, the terms that we chose to base our work on are those that reasonably capture common concepts of interest by us and other works [17].

2.1.2 Functionalities of Attention

Attention can be manifested in different manners depending on the goal. The most notable functionalities shown in intelligent beings are:

- **To select stimuli** such as looking at only a relevant portion of an image – to efficiently use resources on relevant information.
- **To sustain focus** on a specific semantic element for a period to complete a task.
- **To guide processing** in a sequential manner that is relevant for a task.
- **To orient resources** to new important stimuli – such as an abrupt noise coming from somewhere – or even in alternating the focus to multiple tasks at the same time.

2.1.3 Bottom-up and Top-down Attention

Focus may emerge in two fundamentally different manners [7] [10]. In bottom-up Attention, the act of focusing is involuntarily started and guided by (usually) external and conspicuous stimuli, such as a shattering glass that tends to make us immediately turn our heads towards where the noise origin. Another example is visual saliency (Figure 2): a glowing red ball suddenly appearing in your field of vision will probably grab your focus. In top-down Attention, cognition and goals guide the focus voluntarily. If we are talking to someone in a crowded party, for example, we focus on what the specific person is saying – ignoring other people's words – to maintain the conversation.

(a) (b)

Figure 2: Example of visual saliency. b) is the saliency map where higher intensity pixels represent regions that are more salient to humans than original image a).

2.1.4 Soft and Hard Attention

In recent years, there has been a useful distinction between soft and hard Attention [37]. Soft Attention regards defining a continuous distribution of importance across all elements of information for some task. In the example of visual saliency, one can determine a saliency map M to a given image I where each pixel will have a value in $[0, 1]$ regarding its saliency. Hard Attention regards determining a discrete subset of important information elements. Using the problem of visual saliency again as an example, one might want to determine a specific location (i, j) of the image to be used as the center of a small patch of the image that is the most relevant to be further processed.

2.2 Deep Learning

Deep Learning (DL) is a trend in modern AI [21]. Although DL was broadly adopted around 12 years ago, some of its concepts date to much earlier than that [21]: discussions about the foundations of artificial neural networks go back to the 1950s, the introduction of backpropagation to the 1970s and many other vital concepts that are popular mostly in the last decade or less were introduced more than 30 years ago. Many fields of AI witnessed a significant shift in paradigm in the last years: models applying DL concepts now achieve state-of-the-art results in different problems regarding Computer Vision, audio processing, NLP, neural computation, among others [13]. DL used in supervised, unsupervised and reinnforcement learning [21].

One of the critical concepts of DL is that of the hierarchy of features [21]: A deep sequence of layers apply non-linear transformations to the data in such a way that many models learn to extract features of hierarchical levels of abstraction. For this reason, DL is also regarded as Representation Learning. This characteristic enables such models to learn the latent structure in intrinsically unstructured data such as images, text, and audio signals. Another advantage is that of transfer learning: models that are primarily trained for a given task can be used and adapted for another task while using at least part of the representations learned. We discuss some concepts related to DL in the following items.

2.2.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) are usually adopted to predict an output by employing learning a non-linear function approximation. The ideas used in ANNs date to more than 50 years ago [32] and many of them are inspired by observed mechanisms of the human brain. Most of DL models are a variation of one of the families of ANNs that will be briefly discussed here.

One of the most basic examples is that of Multi-Layer Perceptrons (MLPs). The main characteristic of this model is the use of hidden layers and neurons that are a linear combination of previous layers followed by a non-linear activation. Each layer l_k (with n neurons) is connected to the previous layers l_{k-1} (with m neurons) and the neuron l_k^i, $1 \leq i \leq n$ is given value:

$$l_k^i = h\left(\sum_{j=1}^{m} l_{k-1}^j w_k^j + b_k^j\right)$$

Commonly used activation functions are the sigmoid, hyperbolic tangent and the Rectified Linear Unit (ReLU):

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$

ReLU is currently broadly adopted due to its high efficiency and training speed [25].

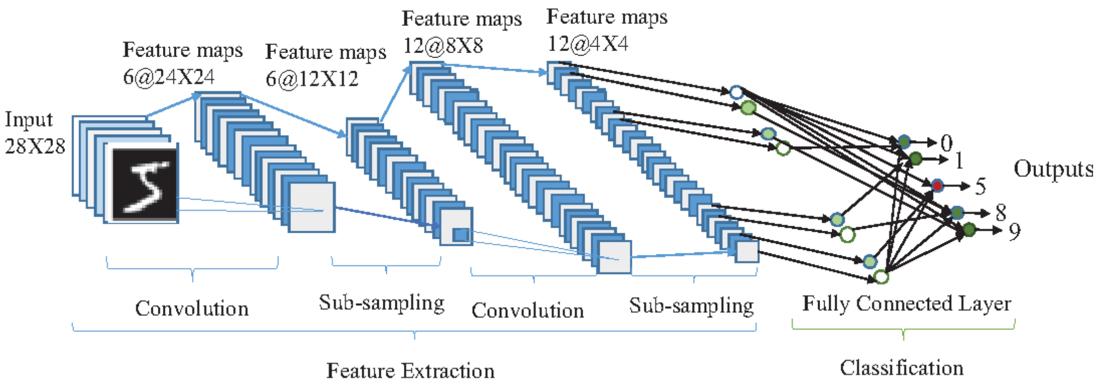

Figure 3: Diagram of a convolutional neural network. Learned filters extract features in an increasingly hierarchical manner.

2.2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are widely used in Computer Vision tasks such as image classification, localization, and semantic segmentation. CNNs use the fact that images tend to have correlated pixels and use convolution filters in an hierarchical manner (Figure 3) to learn features in increasing abstraction. For a certain layer, the i-th feature map m_i is, given filter weights W_i, bias b_i and nonlinearity function $h(x)$, obtained as:

$$m_i = h\left(W_i * x + b_i\right)$$

with $*$ as the convolution operation.

2.2.3 Recurrent Neural Networks

A recursive architecture characterizes recurrent Neural Networks (RNNs) that uses the input of the current step and the output of the previous step to compute the predictions. The hidden state h_t at time step t, given input x_t, weight matrix W, previous state h_{t-1}, hidden-state-to-hidden-state matrix U and non-linearity $f(x)$ is given by:

$$h_t = f\left(Wx_t + Uh_{t-1}\right)$$

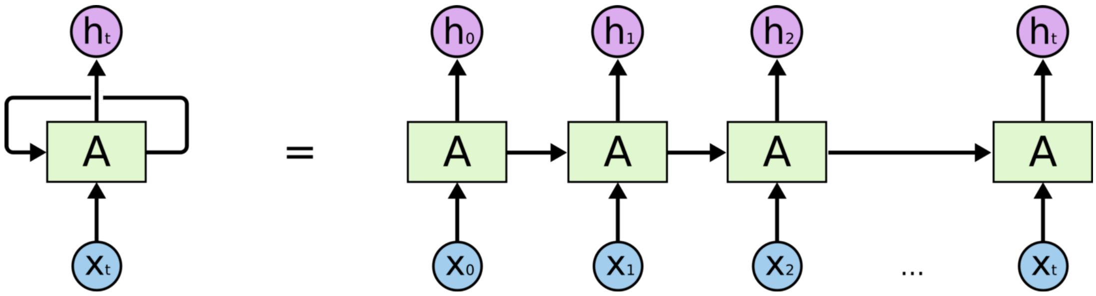

Figure 4: Diagram of a recurrent neural network. Time steps map previous outputs and current input to another time step.

These architectures are widely used in NLP tasks such as machine translation [39]. Some variations over the original underlying architecture such as LSTMs are also broadly adopted.

Modern architectures Recent work on Deep Learning propose models of greater complexity than cited before. Some present completely novice architectures, but most of the work are a combination of underlying neural networks architectures. The final results, however, may consist of entirely new contributions to new tasks. Some examples will be discussed in section 3.

2.2.4 Learning process

The act of learning the appropriate weights of a given model is usually obtained by the minimization of a differentiable loss function that is the cost function $L(y, \hat{y})$ that characterizes the error between the true value y and the predicted value $\hat{y}$. Backpropagation [20] plays an essential role in DL because it is used to adjust the weights θ of models that have a differentiable cost function. A typical training process is composed of a forward-propagation a step which computes the predictions over a set of input samples and a backpropagation step which calculates the loss function and adjusts the weights of the model. In DL, common adjustment methods includes Gradient Descent (GD) [33] or variations which, for a given minibatch, adjusts weights according to:

$$\theta_{i+1} = \theta_i - \alpha \frac{\partial J}{\partial \theta}$$

where α is the learning rate.

3 Related Work

The topic of integrating Attention concepts into Deep Learning has been increasingly frequent in the community [37]. Augmenting the capabilities of neural network architectures with Attention has shown promising results in problems from a variety of fields in which Deep Learning is currently being applied to, such as Computer Vision, Natural Language Processing, and differentiable programming in general. In this subsection, we address some recent works. We highlight how the authors used attention and how it affected the performance of the proposed models on evaluation tasks.

3.1 Attention-based Encoder-Decoder Networks

Encoder-decoder networks are a general framework generally used for mapping from input to outputs that both are of highly-dimensional (often unstructured) data, having been successfully used for tasks such as machine translation [6]. One drawback of such architecture is that the encoded feature vector is of fixed size and structure – regardless of the input – and not necessarily preserves spatial/temporal structure from the data. The work in [5] proposes the usage of an attentional module in between encoder and decoder. The proposed model's encoder produces feature vectors that have an explicit spatiotemporal structure (*context set*) of the input and the attentional module uses a relevance evaluation method to select a subset of the outputs – either by soft Attention or hard Attention. This allows the encoder-decoder for more flexibility to select the components of the input that is of more relevance. The authors implemented and evaluated the method for several applications:

- *Image Caption Generation*: The goal of the task is to provide a natural language description of an input image. The proposed model uses a CNN as encoder and RNN as decoder – with the attentional model in between. The model was ranked third in *MS COCO Captioning Challenge* and provided highly interpretable results regarding the importance of the regions of the image to each component of the sentence (the model is depicted in Figure 1).

- *Neural Machine Translation*: The authors proposed an RNN architecture augmented with the Attention module, which provided a relative improvement of roughly 60% when compared to the same model without Attention. The model also performs better than state of the art in some languages. It was also possible to obtain a weight matrix that maps the importance of input to output words since the context set provides basic information of the input (Figure 5).

- *Neural Speech Recognition*: The goal of the task is to translate audio to text sentences by using fully neural networks. The proposed model uses RNNs between the Attention module, and the model achieved state-of-the-art results in the TIMIT corpus [11] and the outputs provide Attention weights from the input signal to produced phonemes.

Overall, the proposed technique – besides achieving state of the art results – produces a semantic mapping from the input space to the output space even when they are of different nature – without explicitly being supervised to produce this mapping.

3.2 Adaptive computation time for RNNs

Most of the current work uses Attention as a mechanism for filtering. The authors of the work in [14] propose an RNN augmented with an Attention module that allows the dynamic inference of many computation steps for each time step. It uses soft Attention to determine when to stop (Figure 6). The ability to allocate computational resources is an essential function of Attention. The authors show that the mechanism allowed for the model to achieve considerably superior results in tasks such as adding and sorting (when compared to a model without adaptive computation time) because the model was enabled to perform more or fewer operations depending on the dynamic allocation of each step. The authors also tested the model in the problem of character prediction on the Hutter prize Wikipedia dataset [18], in which the model yielded insights into the structure of the input data.

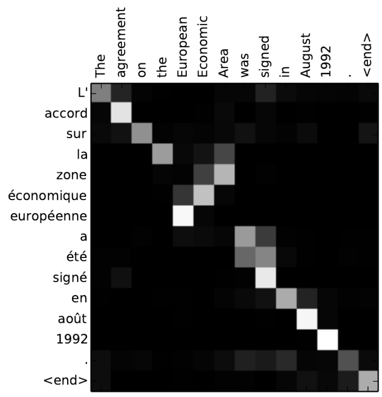

Figure 5: Visualization of Attention weights of the neural machine translation model based on Encoder-Decoder networks. Figure from [5].

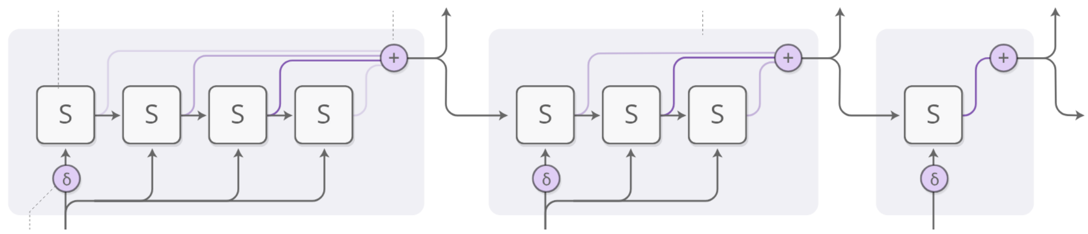

Figure 6: Diagram of the adaptive RNN. Each time step can have the number of computations varied by an Attention distribution. Figure from [27].

3.3 Neural Turing Machines (NTMs)

Neural Turing Machines [15] is one of the first attempts at building models that can learn to formulate programs based on DL architectures with continuous cost functions – and thus trainable via gradient descent. The proposed model is composed of an RNN connected to an external memory bank – which can be read/modified by the use of reading/write heads in the model. The Attention mechanism is the component that allows for the read and writes operations to be differentiable. On every read/write step, there is an Attention distribution – which is updated each step via content-based and location-based methods – that operates on vectors in the memory locations in a continuous manner. The authors show that the model can learn simple algorithms such as sorting and copying sequences.

3.4 Recurrent Attention Model (RAM)

The work [24] considers a commonly known problem in Computer Vision: it is usually expensive to perform processing on images and widely used current models such as CNNs tend to require computational resources proportional to the number of pixels in the image. The work proposes a *Recurrent Attention Model (RAM)*, a recurrent neural network augmented by an attentional component regarded as *Glimpse Module* that is trained via Reinforcement Learning. The Glimpse Module enables the network to select a point in the image from which it extracts glimpses. – patches of the image at different resolutions but with the same dimensions. These glimpses and the selected location are encoded and given as input to produce the new hidden state of the core RNN architecture (Figure 7). The dimensionality of the glimpse is much smaller than that of the image and furthermore

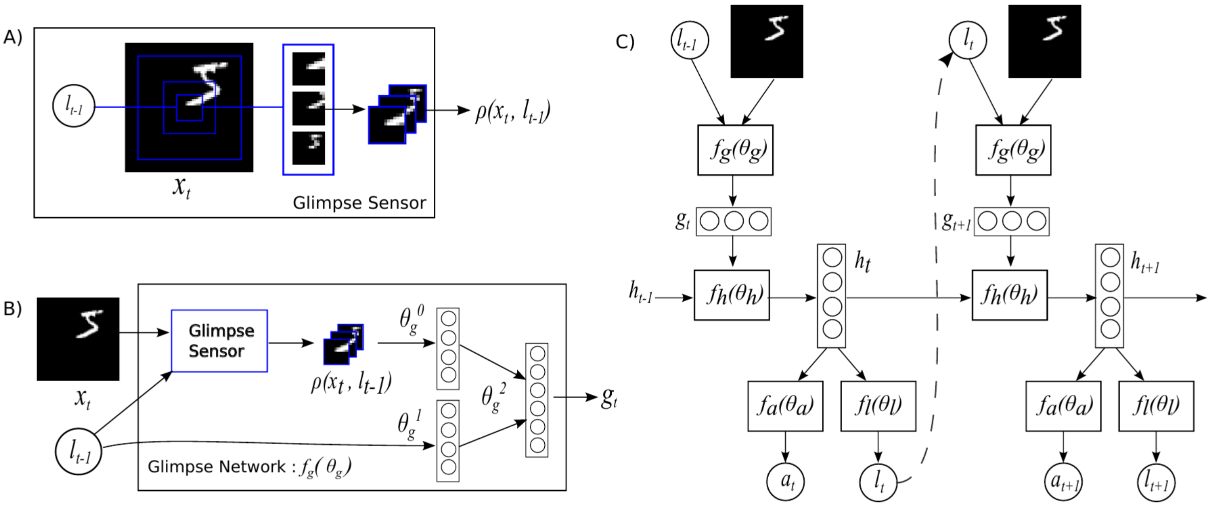

Figure 7: Overall architecture of the recurrent attentive model. **A)** is the *glimpse sensor* that extracts patches of different resolutions from image according to the location being attended. **B)** is the *glimpse module*, which combines information from previous attended locations and glimpses to encode a hidden state. **C)** is the RNN architecture augmented with the glimpse module. Figure from [24].

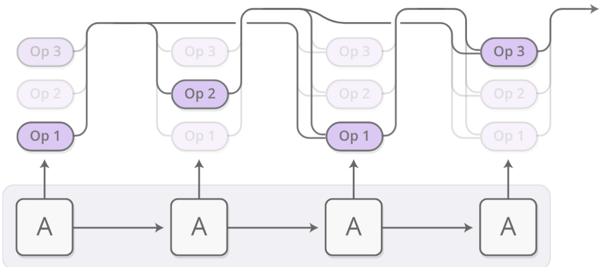

Figure 8: Sequence of operations being carried out by the Neural Programmer model. Attention distributes the weights to each operation. Figure from [27].

does not depend on the dimensions of the input image. The authors evaluate the model for classification tasks in the MNIST [8] dataset and variations in which the input images are filled more background pixels (resulting in a larger image) and clutter. The proposed model outperforms a convolutional neural network baseline. Furthermore, the attentional module in the model enables it to perform the same amount of computation regardless of the input size of the image and to focus sequentially only at the relevant parts of the image, which reduces the adversary effect of clutter.

3.5 Neural Programmer

Deep Learning techniques have been useful for perception tasks in the last years, but tasks that involve complex logic and reasoning are still a major challenge. Authors in [26] propose a model that learns to induce programs by composing basic logic operations into more complex ones in sequence (Figure 8). The model is differentiable and thus trainable via gradient descent because the authors use an Attention distribution at each step to select the operations to be used. The authors of the work evaluate the model on a synthetic table-comprehension dataset. The model achieved nearly perfect accuracy and yielded superior performance compared to LSTMs.

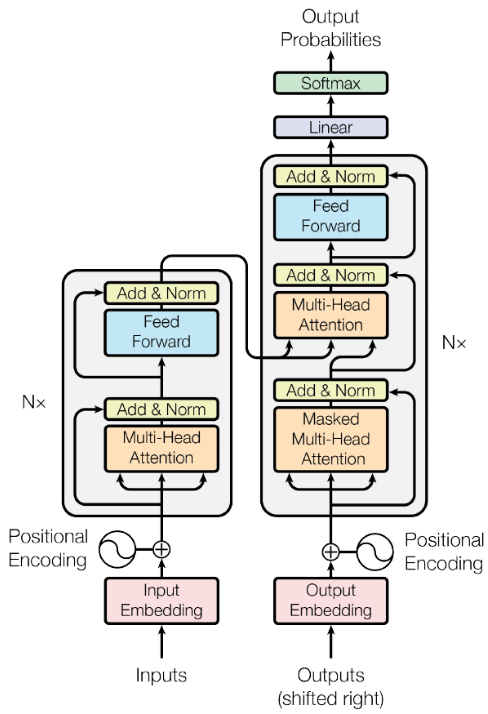

Figure 9: The transformer architecture. Figure from [35].

3.6 The Transformer architecture

For sequence modeling tasks such as machine translation, there is an extensive use of sequence models such as RNNs and more specifically LSTMs. The work proposed in [35] aims at overcoming some challenges inherent to such sequential models, such as performance and obstacles to applying parallelization to the process. Complications also arise when the content of the inputs are long (such as long sentences in the text) and recurrent models present difficulties in establishing relationships between words. The authors present the *Transformer*, an encoder-decoder feed-forward architecture with Attention as a critical element. The input sentences are embedded, and positional encoding is applied. Then, each layer of the transformer and the decoder employ either *scaled dot-product Attention* or *multi-head Attention*, which allows for contextual mapping and representation of long-relationships. The proposed model achieved state of the art results on *WMT 2014 English-to-German* and *WMT 2014 English-to-French* translation tasks.

4 Methodology

To achieve our goals, we will follow the methods described next.

4.1 Activities

The work can be summarized in three main activities (or **phases**):

- **1. Literature Review**: an extensive survey on current uses of Attention in modern Deep Learning.
- **2. Proposal of an Attention framework for Deep Learning**: defining a component set of Attention elements currently used in Deep Learning design from results of the previous phase and further survey.

- **3. Implementation and validation of Attention in Deep Learning**: proposing a model with components of Attention and evaluating it on a set of tasks.

More specifically, the activities to be executed are:

- **A1.1 - Theoretical definition of Attention and its components:** From a variety of previous works [17][7], we establish a theoretical framework of Attention – extending what was discussed in subsection 2.1 – on which all later work will be based. It is worth noting that this theoretical framework is not necessarily the same as the framework we propose to produce specifically for Deep Learning in phase 2. It will work as the base that will be used to organize and classify the current literature.
- **A1.2 - Elaboration of survey:** Exploration of selected work under the point of view of the theoretical framework established in **A1.1**. For each work, we identify the main components of Attention the authors use, the consequences for the performance in the application domain and elaborate a critical evaluation.
- **A1.3 - Survey article writing:** Writing of an article with results of phase 2 to be sent to an appropriate journal (ACM Computing Surveys).
- **A2.1 - Establishment of Attention components for specific Deep Learning domains:** From the theoretical framework obtained in **A1.1** and the exploration of current uses and results in **A1.2**, we devise sets of useful components of Attention for specific main problem domains in which Deep Learning is broadly used, such as image classification, text-to-speech, language translation, image segmentation.
- **A2.2 - Establishment of Attention framework for Deep Learning:** From the theoretical framework obtained in **A1.1**, exploration of current uses and results in **A1.2** and results from **A2.1**, we elaborate a set of components of Attention under a single framework to be applied to more general areas of use of Deep Learning, such as Computer Vision, Sequence Processing, Program Composition.
- **A3.1 - Arrangement of experiments:** From the framework obtained in phase 2, we select a set of problem domains (such as text-to-speech), Deep Learning models to use, components of Attention to implement and metrics to evaluate the task. The activity aims at selecting all main devised components from phase 2 in order to evaluate the real consequences of their adoption against what was predicted.
- **A3.2 - Execution of experiments:** We implement and execute the planned experiments following a pre-defined protocol that pays particular attention to reproducibility.
- **A3.3 - Evaluation of experimental results:** We evaluate the results using established metrics for each experiment, elaborating discussions that include exciting aspects of the results in general and comparisons between the theoretical predictions and tangible outcomes. It is worth noting that the metrics we'll use will vary depending on the specific problem, but they will always be selected to reflect the improvement of the models with the use of attention.
- **A3.4 - Experiments article writing:** Writing of an article with results of phase 3 to submit for publication.

Other activities to be done related to the masters program are:

- **A0.1 - Course's requirement fulfillment.**
- **A0.2 - Qualification Exam.**
- **A0.3 - Masters dissertation.**
- **A0.4 - Defense of masters dissertation.**

4.1.1 Schedule

Table 1 details the project schedule. Considering the start of the program in the second semester of 2018, the duration of the work is expected to be one and a half years.

Table 1: Project schedule.

Activity	**2018**			**2019**											
	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
A0.1	*	*	*												
A1.1	*														
A1.2				*	*	*	*	*	*	*					
A0.2								*							
A1.3										*					
A2.1										*					
A2.2											*				
A3.1											*				
A3.2											*	*	*	*	
A3.3														*	
A3.4														*	
A0.3													*	*	*
A0.4															*

5 Theoretical Framework for Attention

Activity **A0.1** (described in 4.1) was successfully completed, resulting in a theoretical framework [31] for attention upon which we can build future work. This task is essential since we want to be as clear as possible regarding what we define as attention. In the framework, we define a set of *entities of interest* and the phenomenon of Attention in terms of its *functionalities* and how it relates to the entities. An initial definition was given in 2.1.1 – we used part of it as a basis for the complete framework. Here we mention the most relevant aspects of the framework.

We now briefly discuss our formulated theoretical framework for the concept of Attention. This framework consists of two main parts:

- A *definition* of Attention in terms of its functionalities;
- A *model* of Attention.

Note that the first element aims at answering the question "*What* is Attention?" while the second aims at explaining *how* Attention emerges.

5.1 A definition of Attention

In this definition, we define a set of *entities of interest* and the phenomenon of Attention in terms of its *functionalities* and how it relates to the entities.

We believe our definition encompasses what we generally (and intuitively) refer to as attention while being not too broad. Also, the definition is given in terms related to Computer Science, so it's functionality translates to the domain, which is essential since we (so far) intend to develop AI using computers as we know it today. We may not encompass every aspect of Attention and even be conflicting with other definitions. However, this is the set of postulates that we think is the most precise and useful and thus this is what we choose to use for future work to be based on.

5.1.1 Entities

Below is the list of entities - orterms - we use in this work, along with a brief discussion of the meaning we give to each term in the context of this work.

- **Data:** information, stimuli. It may be internal or external. Examples: visual information, audio, memories.
- **Program:** algorithm, sequence of computer (or mental) operations. Programs use data as input to carry out a sequence of operations that produces output data and/or actions.
- **Process:** the execution of a program on a specific data instance.
- **Computer:** the executor of processes, the brain.
- **Resource:** when not specified, we mean computational resources, e.g., CPU time.
- **Time:** the flow of time.
- **World:** the external environment.
- **Agent:** the actor in the world.
- **Actions:** the interaction of the agent with the world.
- **Goals:** the ends, objectives to be met.

5.1.2 What is Attention?

Data, *programs* and *processes* are virtually *infinite*. Computational *resources* and *actions* are finite. *Attention* is *the system of allocating resources to processes*. In other words, *attention* is the entity in *agent* that, given *context* and a set of *processes*, *allocates resources* to execute each of them in order to *produce outputs* in form of *data* and *actions* in a *correct sequential manner* and in *sensible time* in order to reach *goals*.

5.2 A model of Attention

We propose a model for the phenomenon of Attention. Following the definition given in subsection 5.1, in our model we assume that Attention takes place in the context of a mind that behaves like a computer that executes processes: it processes inputs via algorithms to produce an output in discrete steps. We propose that *Attention can emerge in any process* to be executed by such mind employing *a series of components* – which we call *attentional modules*. These modules can alter data being processed and the execution flow of the algorithm and provide the functionalities of Attention.

Figure 10: Attentional module.

Figure 10 illustrates the attentional module. At each time step t, the module receives as *input*:

- Current *outer state* $o_t \in O$, where O is the *outer state set.*
- Group of *focus targets* $\tau_t = \{\tau_{t1}, \ldots, \tau_{tk}\}, \tau_{ti} \in T$, where T is the *focus target set.*
- Past *inner state* $\iota_{t-1} \in I$, where I is the *inner state set.*

The module produces as *output* (as a function of both inputs):

- Current *inner state* $\iota_t \in I$.
- Current *focus output* $\alpha_t = \{\alpha_{t1}, \ldots, \alpha_{tk}\}, \alpha_{ti} \in A$, where A is the *focus output set.*

5.2.1 Focus output

The focus output is the main element of the module: it can be used to allocate *finite resources* to a set of candidate targets by giving them an importance score which can be used in any arbitrary way in following steps – such as choosing the amount of computation to be dedicated to an element or which elements will be used as input to another stage. Each element α_{tk} is respective to a target element τ_{tk}. Target elements ($\tau \in T$) may effectively be *programs* (tasks) or *data.*

Soft and Hard Attention: The focus output will generally be such that it acts as either *Soft* or *Hard Attention*:

- **Soft Attention:** $A = [0, 1]$, with $0 \leq \sum_{i=1}^{k} \alpha_{ti} \leq 1$
- **Hard Attention:** $A = \{0, 1\}$, with $0 \leq \sum_{i=1}^{k} \alpha_{ti} \leq M$ and $0 \leq M \leq |\tau_t|$

Using the output of the focus function: The focus function output may be used for the allocation of some resource in various ways, such as:

- Choosing the **amount** of **computation time** to be used at a certain step;
- Choosing a **subset** of **elements** to carry out further computations;
- **Weighting elements** to perform a certain computation.

5.2.2 Modules forming an attentional system

A system with Attention may contain more than one attentional module – even in a recursive manner. Together, these modules always perform the function to allocate resources to processes.

Figure 11 shows the diagram of a possible system with attention. The module *TaskATT* uses hard attention to select a certain task k to be executed for some time at time step t. Among the computations of task k, there is the module *DataATT*, uses soft attention to allocate resources to a set of items. It is worth noting that time is relative to each attentional module: *TaskATT* has a temporal course over time steps t that is different from that of *DataATT*, which is over time steps t'. Also, their sets of inputs and outputs may differ.

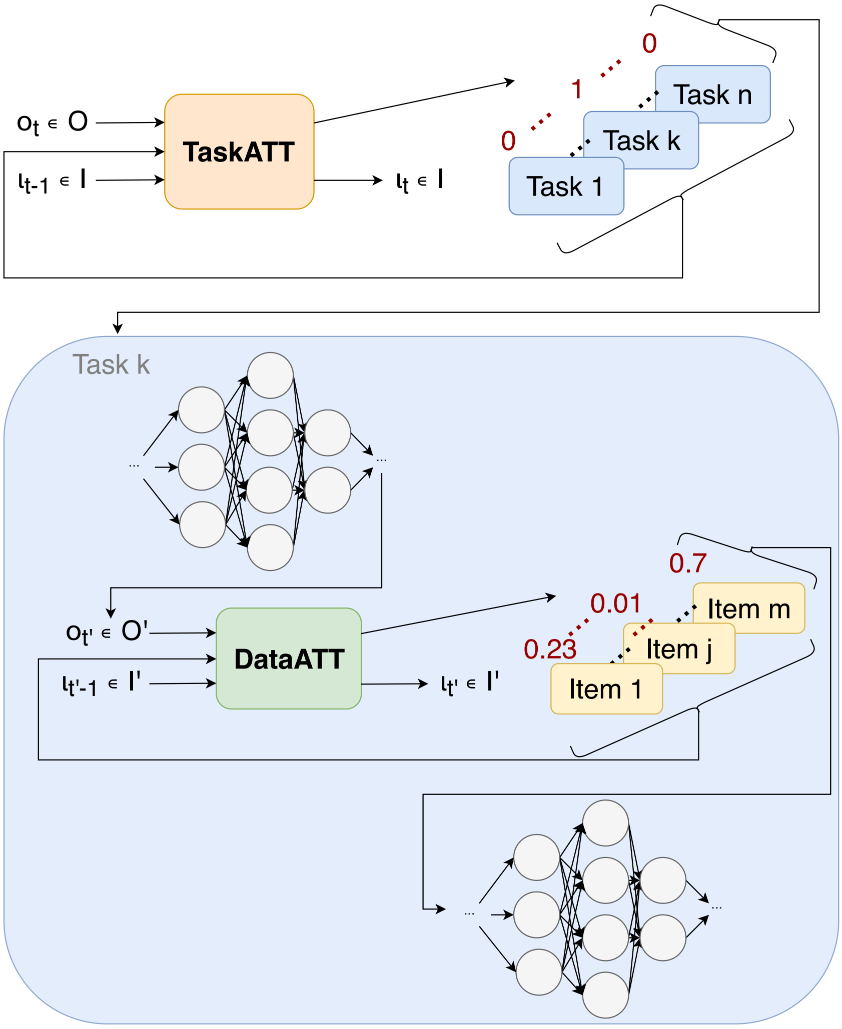

Figure 11: Example of a system that uses attention.

5.3 Validating the framework

In this subsection, we investigate some recent work on attention to evaluate how they would fit in our framework.

5.3.1 Image Caption Generation

The work described in [38] is among the first to propose using attention to image caption generation: the encoding of the input image is represented as a set of vectors – each respective to a certain spatial region of the image – and the attentional component gives weights to each vector at each step to produce another vector to be used in further computations.

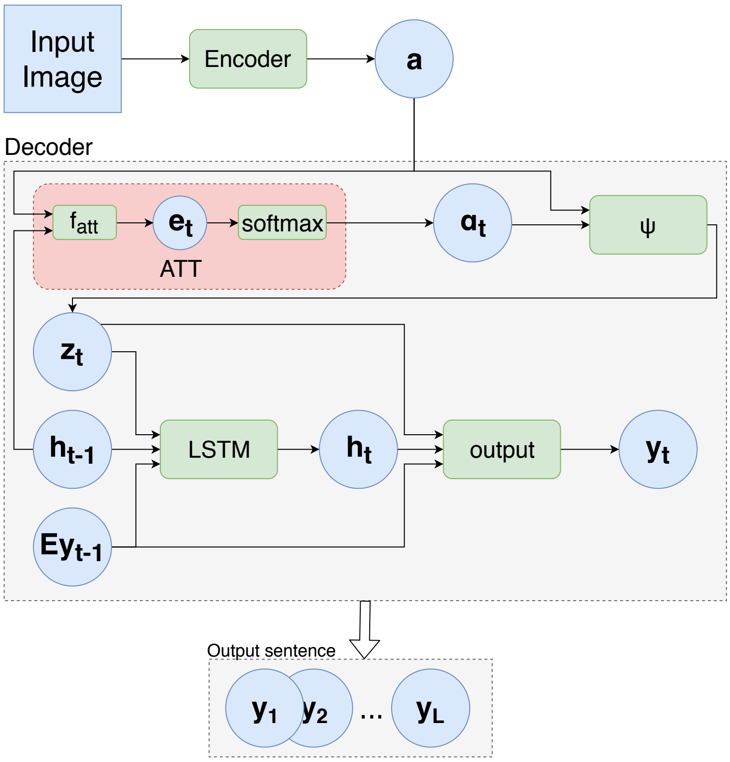

Figure 12: Model proposed for image captioning in [38] with attentional module.

Figure 12 illustrates the model proposed in this work. The steps that calculate the attention to each encoding vector can be encapsulated as an attentional module under our modeling: a, the input image encoding, is the *focus target input* τ_t, h_{t-1}, the hidden state of the model LSTM, is the *outer state input* o_t and α_t, the weights given to each encoding vector, is the *focus output*. In this case, $A = [0, 1]$. Note that, in this case, the *internal state* is empty.

5.3.2 Adaptive Computation Time

The work [14] proposes an RNN that can perform a variable number of computation sub-steps for each time step t'. The main idea is to calculate an amount $0 \leq p_{t',t} \leq 1$ to be spend for each computation sub-step t up until the moment the total spent reaches the budget of 1 (in which moment the computation is halted). The final value $y_{t'}$ is computed as an weighted average of the intermediate $y_{t',t}$ values and the weights are the values $p_{t',t}$.

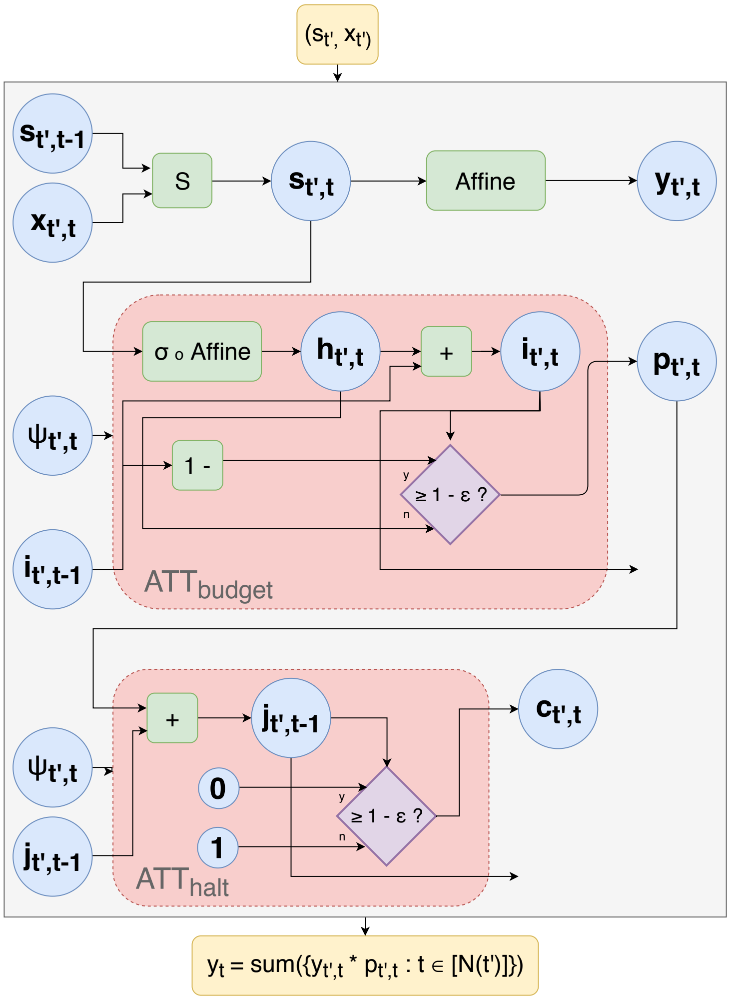

Figure 13: Model proposed for image captioning in [14] with attentional module.

Figure 13 illustrates the model proposed in the work. The proposed model can be thought of as having two attention modules:

- ATT_{budget}, which computes the value $0 \leq p_{t',t} \leq 1$ to be spent at a given sub-step. In this analogy, $s_{t',t}$ – the state of the RNN cell – is the *outer state* o_t; ψ_t – a dummy element representing the current computation sub-step – is the *target* τ_t; and $i_{t',t}$ is the *inner state*. The *focus output* $p_{t',t}$, besides representing values to be consumed from the budget, can be thought of as an importance weight for the final output y_t, since the produced values are used to computed the weighted average.

- ATT_{halt}, which computes the value $c_{t',t} \in \{0, 1\}$, which is 1 if the cell should continue further sub-steps and 0 otherwise. In this analogy, $p_{t',t}$ is the *outer state* o_t; ψ_t – a dummy element representing the current computation sub-step – is the *target* τ_t; and $j_{t',t}$ is the *inner state*.

It is interesting to note that an effect that emerges from these two blocks is that *the model can allocate resources to processes* both by *choosing the data to use* (in the computation of each y_t weighted by a focused output) and *choosing the amount of computation time to use*.

5.3.3 Recurrent Attention Model of Visual Attention

The work [24] proposes a general recurrent model that uses visual attention at each step by selecting a retina-like representation of a portion of the input image to carry out further computations. At each time step t, the model uses the selected location l_{t-1} to extract a retina-like representation from input image. An arbitrary action a_t can be executed to possibly alter the environment.

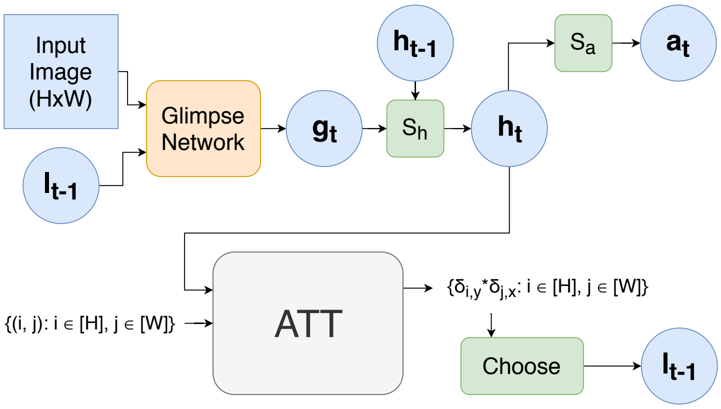

Figure 14: General recurrent architecture proposed in **[ref:ram]** with attentional module.

Figure 14 illustrates the model proposed in the work. In this representation, the hidden state of the RNN h_t is the *outer state* input o_t; The set of possible pixel coordinates $\{(i, j) : i \in [H], j \in [W]\}$ (with H, W as the height, width of the image) is the *focus targets* input τ_t; and the set $\{\delta_{i,y}\delta_{j,x} : i \in [H], j \in [W]\}$ is the *focus output*. Note that only the element $\delta_{i,y}\delta_{j,x}$ – which is respective to the chosen pixel coordinates (x, y) is equal to 1.

6 Survey

Activity **A1.2** (described in 4.1) is currently in progress. We now report the progress made so far.

6.1 Collection of relevant works

The first step was to search in the literature for works which applied some form of "attention" to Deep Learning. A methodology for searching and selecting the works was established, which we summarize below:

- **Publication date range**: from 2014 to 2019
- **Databases searched**:
  - **arXiv** - `https://arxiv.org/`
  - **DeepMind** - `https://deepmind.com/research/publications/`
  - **Google AI** - `https://ai.google/research/pubs/`
  - **OpenAI** - `https://openai.com/research/#publications`
  - **Facebook AI research** - `https://research.fb.com/publications/`
  - **Microsoft research** - `https://www.microsoft.com/en-us/research/search/`
  - **Amazon research** - `https://www.aboutamazon.com/publications`
  - **DBLP** - `https://dblp1.uni-trier.de`
  - **NIPS** - `https://nips.cc/`

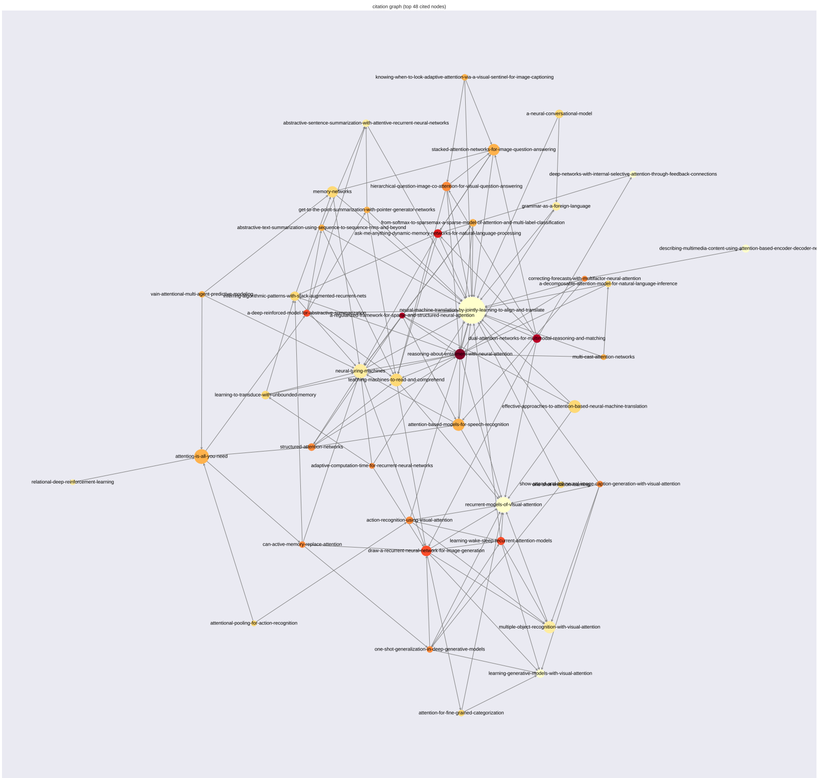

Figure 15: Citation graph of the top cited papers generated by litrev tool.

- **ICML** - `https://icml.cc/`
- **ICLR** - `https://iclr.cc/`
- **AAAI** - `http://www.aaai.org/`
- **CVPR** - `http://cvpr2018.thecvf.com/`
- **ICCV** - `http://iccv2017.thecvf.com/`

- **Terms (in title or abstract)**: “attention”, “attentive” or “attentional”

The relevance of each work was confirmed upon the reading of the abstract. As a result, we collected around 300 papers.

6.2 Literature Visualizations

Some visualizations were generated for the works collected with the intent to have some insights regarding the collection acquired. Some visualizations generated were: Citations graph (authors and works), Abstract/title word frequencies, Frequency of attention-themed papers over the years. Relative Frequency of attention-themed papers over years for some databases (arXiv, CVPR, DBLP). Visualizations were generated by a set of scripts which can be found at `https://github.com/erikperillo/litrev`. One example of visualization is figure 15.

6.3 Paper relevance analysis

With the collection in hands, the next goal was to assess the relevance and problem domain of each work via a paper relevance analysis [30]. To each work, we attributed the citation count, domains (e.g., Computer Vision), subdomains (e.g., image classification), and an impact score ranging from 1 to 5. This score was assessed in a quick and rough manner via the abstract of each work, and it was given based on criteria such as:

- How innovative is the proposed model(s) of the work?
- How general is the proposed model(s)?
- Does the proposed model(s) archives/surpasses state-of-the-art in some task?
- Is attention a central component to the results of the work?

6.4 Reading and summarization of works

This step (which is in progress) aims at reading each paper of the collection in depth (in order of relevance, from highest to lowest) and generating a summary for each work. A summary template was formulated [29] and summaries were generated for some works. This is the main step of the survey, and it is expected to be the longest. The goal is to further refine our theoretical framework and to guide the reading of future papers as we read those papers – in an iterative manner. Survey has shown so far that the use of attention in Deep Learning has indeed provided improvements in basically all subfields of DL. Table 2 shows the taxonomy classification for some of the most important works analyzed so far in the survey.

Table 2: Works classified by taxonomy

Model	Selection Target	Continuity
[38]	Data	Soft
[24]	Data	Hard
[9]	Data	Soft
[14]	Resources	Soft
[15]	Data	Soft
[4]	Data	Soft
[26]	Program	Soft
[35]	Data	Soft

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". In: *CoRR* abs/1409.0473 (2014). arXiv: 1409.0473. URL: http://arxiv.org/abs/1409.0473.

[2] Yoshua Bengio et al. "Curriculum learning". en. In: *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*. Montreal, Quebec, Canada: ACM Press, 2009, pp. 1–8. ISBN: 978-1-60558-516-1. DOI: 10.1145/1553374.1553380. URL: http://portal.acm.org/citation.cfm?doid=1553374.1553380 (visited on 09/21/2018).

[3] William Chan et al. "Listen, Attend and Spell". In: *CoRR* abs/1508.01211 (2015). arXiv: 1508.01211. URL: http://arxiv.org/abs/1508.01211.

[4] KyungHyun Cho, Aaron C. Courville, and Yoshua Bengio. "Describing Multimedia Content using Attention-based Encoder-Decoder Networks". In: *CoRR* abs/1507.01053 (2015). arXiv: 1507.01053. URL: http://arxiv.org/abs/1507.01053.

[5] Kyunghyun Cho, Aaron Courville, and Yoshua Bengio. "Describing Multimedia Content using Attention-based Encoder–Decoder Networks". In: *IEEE Transactions on Multimedia* 17.11 (Nov. 2015), pp. 1875–1886. ISSN: 1520-9210, 1941-0077. DOI: 10.1109/TMM.2015.2477044. arXiv: 1507.01053. URL: http://arxiv.org/abs/1507.01053 (visited on 09/11/2018).

[6] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *arXiv:1406.1078 [cs, stat]* (June 2014). arXiv: 1406.1078. URL: http://arxiv.org/abs/1406.1078 (visited on 09/21/2018).

[7] E.L. Colombini, A. da Silva Simoes, and C.H. Costa Ribeiro. "An Attentional Model for Autonomous Mobile Robots". In: *IEEE Systems* 99 (2016), pp. 1–12.

[8] L. Deng. "The MNIST Database of Handwritten Digit Images for Machine Learning Research". In: *IEEE Signal Processing Magazine* 29.6 (Nov. 2012), pp. 141–142. ISSN: 1053-5888. DOI: 10.1109/MSP.2012.2211477.

[9] Yan Duan et al. "One-Shot Imitation Learning". In: *CoRR* abs/1703.07326 (2017). arXiv: 1703.07326. URL: http://arxiv.org/abs/1703.07326.

[10] Simone Frintrop. "VOCUS: a visual attention system for object detection and goal-directed search". In: *IN LECTURE NOTES IN ARTIFICIAL INTELLIGENCE (LNAI)*. Springer, 2005.

[11] J. S. Garofolo et al. *DARPA TIMIT Acoustic Phonetic Continuous Speech Corpus CDROM*. 1993.

[12] Ian J. Goodfellow et al. "Generative Adversarial Networks". In: *arXiv:1406.2661 [cs, stat]* (June 2014). arXiv: 1406.2661. URL: http://arxiv.org/abs/1406.2661 (visited on 09/21/2018).

[13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL: http://www.deeplearningbook.org.

[14] Alex Graves. "Adaptive Computation Time for Recurrent Neural Networks". en. In: *arXiv:1603.08983 [cs]* (Mar. 2016). arXiv: 1603.08983. URL: http://arxiv.org/abs/1603.08983 (visited on 09/11/2018).

[15] Alex Graves, Greg Wayne, and Ivo Danihelka. "Neural Turing Machines". In: *CoRR* abs/1410.5401 (2014). arXiv: 1410.5401. URL: http://arxiv.org/abs/1410.5401.

[16] Kaiming He et al. "Mask R-CNN". In: *CoRR* abs/1703.06870 (2017). arXiv: 1703.06870. URL: http://arxiv.org/abs/1703.06870.

[17] Helgi Helgason. "General Attention Mechanism for Artificial Intelligence Systems". In: (May 2013).

[18] Marcus Hutter. *The human knowledge compression prize*. URL: http://prize.hutter1.net (visited on 09/21/2018).

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., 2012, pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[20] Yann LeCun. *A Theoretical Framework for Back-Propagation*. 1988.

[21] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". en. In: *Nature* 521.7553 (May 2015), pp. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539. URL: https://www.nature.com/articles/nature14539 (visited on 09/21/2018).

[22] Shane Legg and Marcus Hutter. "Universal Intelligence: A Definition of Machine Intelligence". In: *CoRR* abs/0712.3329 (2007). arXiv: 0712.3329. URL: http://arxiv.org/abs/0712.3329.

[23] Tomas Mikolov, Armand Joulin, and Marco Baroni. "A Roadmap towards Machine Intelligence". In: *CoRR* abs/1511.08130 (2015). arXiv: 1511.08130. URL: http://arxiv.org/abs/1511.08130.

[24] Volodymyr Mnih et al. "Recurrent Models of Visual Attention". In: *arXiv:1406.6247 [cs, stat]* (June 24, 2014). arXiv: 1406.6247. URL: http://arxiv.org/abs/1406.6247 (visited on 09/11/2018).

[25] Vinod Nair and Geoffrey E. Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML'10. USA: Omnipress, 2010, pp. 807–814. ISBN: 978-1-60558-907-7. URL: http://dl.acm.org/citation.cfm?id=3104322.3104425 (visited on 09/21/2018).

[26] Arvind Neelakantan, Quoc V. Le, and Ilya Sutskever. "Neural Programmer: Inducing Latent Programs with Gradient Descent". en. In: *arXiv:1511.04834 [cs, stat]* (Nov. 2015). arXiv: 1511.04834. URL: http://arxiv.org/abs/1511.04834 (visited on 09/11/2018).

[27] Chris Olah and Shan Carter. "Attention and Augmented Recurrent Neural Networks". In: *Distill* (2016). DOI: 10.23915/distill.00001. URL: http://distill.pub/2016/augmented-rnns.

[28] Aäron van den Oord et al. "WaveNet: A Generative Model for Raw Audio". In: *CoRR* abs/1609.03499 (2016). arXiv: 1609.03499. URL: http://arxiv.org/abs/1609.03499.

[29] Erik Perillo and Esther Colombini. *Paper analysis template*. URL: https://drive.google.com/open?id=1jIbHUoVNnxL61DTelpjwZAhNm5HhOOa_YSZCoPqiJxY.

[30] Erik Perillo and Esther Colombini. *Paper relevance analysis*. URL: https://docs.google.com/spreadsheets/d/1VlNJNr0-ujvYQCdMwnMGTOlXLAJhOtzTy4ODbdX9MNg/edit?usp=sharing.

[31] Erik Perillo and Esther Colombini. *Theoretical framework of attention*. URL: https://drive.google.com/open?id=1gK8MjANmC2lQNHmP28-HWbT4giHVU9y5O3nNJioNgjg.

[32] F. Rosenblatt. "The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain". In: *Psychological Review* (1958), pp. 65–386.

[33] Sebastian Ruder. "An overview of gradient descent optimization algorithms". In: *arXiv:1609.04747 [cs]* (Sept. 2016). arXiv: 1609.04747. URL: http://arxiv.org/abs/1609.04747 (visited on 09/21/2018).

[34] Alan M. Turing. "Computing Machinery and Intelligence". In: *Mind* (1950).

[35] Ashish Vaswani et al. "Attention Is All You Need". en. In: *arXiv:1706.03762 [cs]* (June 2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762 (visited on 09/11/2018).

[36] Ashish Vaswani et al. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

[37] Feng Wang and David M. J. Tax. "Survey on the attention based RNN model and its applications in computer vision". In: *CoRR* abs/1601.06823 (2016). arXiv: 1601.06823. URL: http://arxiv.org/abs/1601.06823.

[38] Kelvin Xu et al. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention". In: *CoRR* (2015). URL: http://arxiv.org/abs/1502.03044.

[39] Tom Young et al. "Recent Trends in Deep Learning Based Natural Language Processing". In: *arXiv:1708.02709 [cs]* (Aug. 2017). arXiv: 1708.02709. URL: http://arxiv.org/abs/1708.02709 (visited on 09/21/2018).