A theoretical framework for Attention

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

State University of Campinas

September 4, 2019

1 Introduction

In this document, we briefly formulate a 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.

1.1 Entities

Below is the list of entities — or "terms" — 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 in order 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.

2 What is Attention?

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

2.1 Why is our definition good?

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 its functionality nicely translates to the domain, which is important 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.

2.2 A broad definition of 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 *agents* 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*.

3 How does Attention happen?

The *allocation* of *resources* to *processes* can be thought of as the process of performing *selection* over the course of *time* given a *finite set of options* to be chosen from and a *limited* quantity of *choices* in the context of a *task*.

Let's describe the process in the context of an agent that contains some *computing unit* that executes *processes* (i.e. *programs* processing some *data*). The agent has inerent *constraints*: Limited amount of *computing resources*, *data bandwidth* and *actions*. However, it still has *goals* that should be achieved via the execution of *tasks* in *sensible time*. In such context, the attention mechanism can be responsible for, given *possible choices* of *what to do* and *which stimuli to use*, help perform the appropriate *selection* along time. Attention can help select which programs to be run. Once the programs that will be run are established, attention can also select which subset of the *data* should be used by each program. Note that the way selection is performed over *the course of time* is important for most tasks — this is also covered by attention.

3.1 Attentional Modules

We propose that it is possible to model the selection process of Attention by means of *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 emergence of Attention.

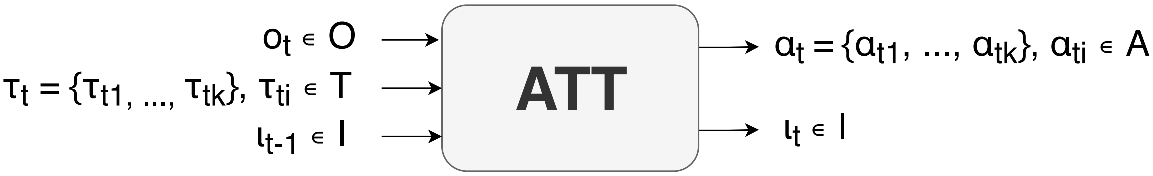
$o_t \in O$
$\tau_t = \{\tau_{t1}, ..., \tau_{tk}\}, \tau_{ti} \in T$
$\iota_{t-1} \in I$
ATT
$\alpha_t = \{\alpha_{t1}, ..., \alpha_{tk}\}, \alpha_{ti} \in A$
$\iota_t \in I$

Figure 1: Attentional module.

Figure 3.1 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*.

The focus target elements are finite and express the idea of a set of possible choices for the selection process. The state elements (both outer and inner) express the fact that selection if often contextual.

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. Each element α_{tk} is respective to a target element τ_{tk}.

A system with Attention may contain more than one attentional modules — even in a recursive manner. Together, these modules always perform the function to provide selection capabilities. The modules can assume different profiles according to their target-focus mapping and the nature of the data involved in the processing. Some of such profiles are worth mentioning and are summarized in Figure 3.1.2.

3.1.1 Soft and Hard selection

The selection may occur either by choosing a discrete subset of the possible choices or by perfoming a "soft" and/or non-deterministic selection, giving real-valued scores to the possible choices. The different types of selection can be implemented with modules by appropriately choosing the focus output set A:

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

3.1.2 Selection state profile

As mentioned before, time is usually important in the context of a task and so attention may provide selection and a specific evolution of selection along the course of time so as to be useful for the task. In some cases, however, selection takes place in a short window so that the past must not be taken into consideration. We can divide both profiles of time signature as *stateful* and *stateless* selection. The presence of inner/outer states (or lack thereof) in the module can implement this behavior:

- **Stateful selection**: $\iota_t \neq \emptyset$ or $o_t \neq \emptyset$
- **Stateless selection**: $\iota_t = \emptyset$ and $o_t = \emptyset$

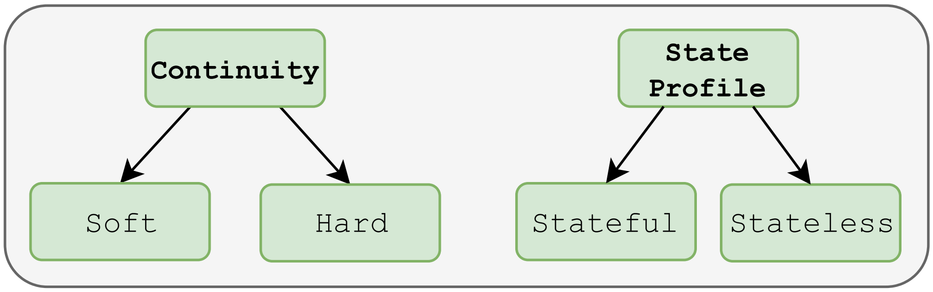

Figure 2: Some of the taxonomy of the selection provided by attentional modules.

3.2 The semantics of selection

Although attention can be simply modeled as process of selecting items from a set of possible choices (Section 3.1), very often there is *semantic meaning* to it – which is given in the *context of the task* being performed. We can cite as an example the act of visually tracking a specific car on the road. Even though attention is technically the simple selection of items from the image (pixels), the semaning meaning is that of *focusing on the car*, which is implemented by proper selection of pixels that refer to the car over the course of time relevant to the task. In this section, we discuss in detail some aspects of the semantic meanings of selection. Figure 3.2.2 summarizes the categorization.

3.2.1 Subjects of selection

The items to be selected (i.e. the target type) may be semantically classified as either *data* or *programs*. Selection of *data* regards selecting targets that are to be used as input to other processes. Selection of *programs* regards selecting targets that refer to programs to be executed (Figure 3.2.1).

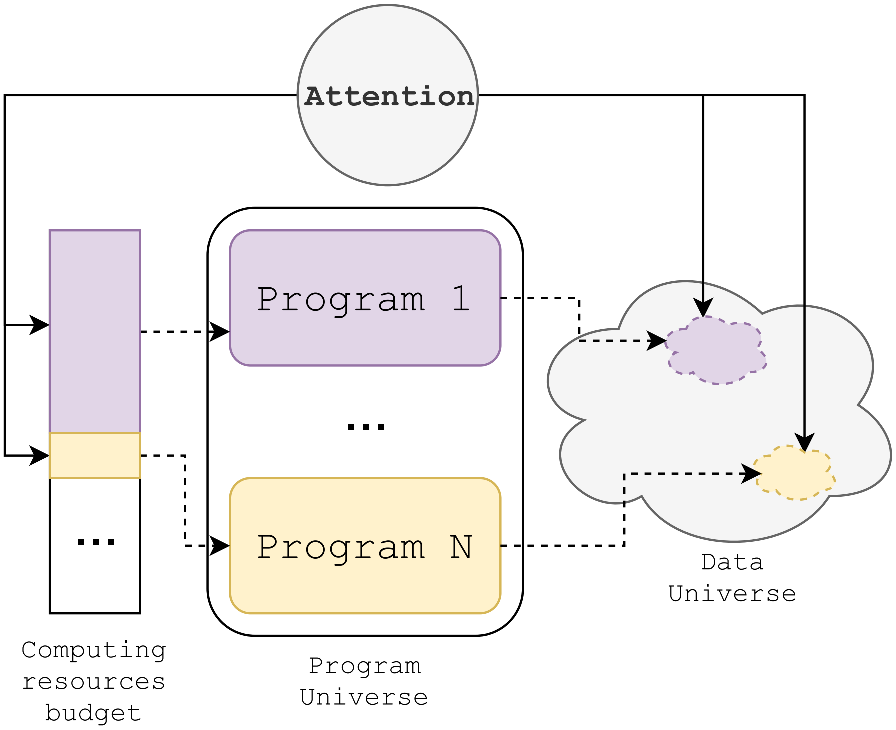

Figure 3: High level overview of the semantic subjects of selection. Given computing time budget and context of the task, the appropriate programs can be allocated and the relevant data can be used by the programs.

We can further classify the semantic meaning of selection of data as:

- **Location-based**: based on the location of the stimuli, such as the coordinates of a pixel.
- **Feature-based**: based on the features of a stimuli, such as color, orientation.
- **Object-based**: based on semantic elements of the stimuli, such as a specific toy in an image.

Figure 3.2.1 shows some examples of each category. An example of a process of selection of data is the selection of a subregion of the input image to be processed in an image classification task. In this case, selection can be considered to be location-based. It is worth noting that, in the end, every data selection can be considered to be location-based. After all, the targets of selection will always be discrete items of a vector (pixels of an image, for example). This further categorization is nonetheless useful.

The semantic meaning of selection of programs can also be further classified into two broad categories:

- **Program choice**: given options choose one (or more) programs to be executed next.
- **Computing time choice**: given computational time "budget" and programs, choose how much computing time to allocate to each program.

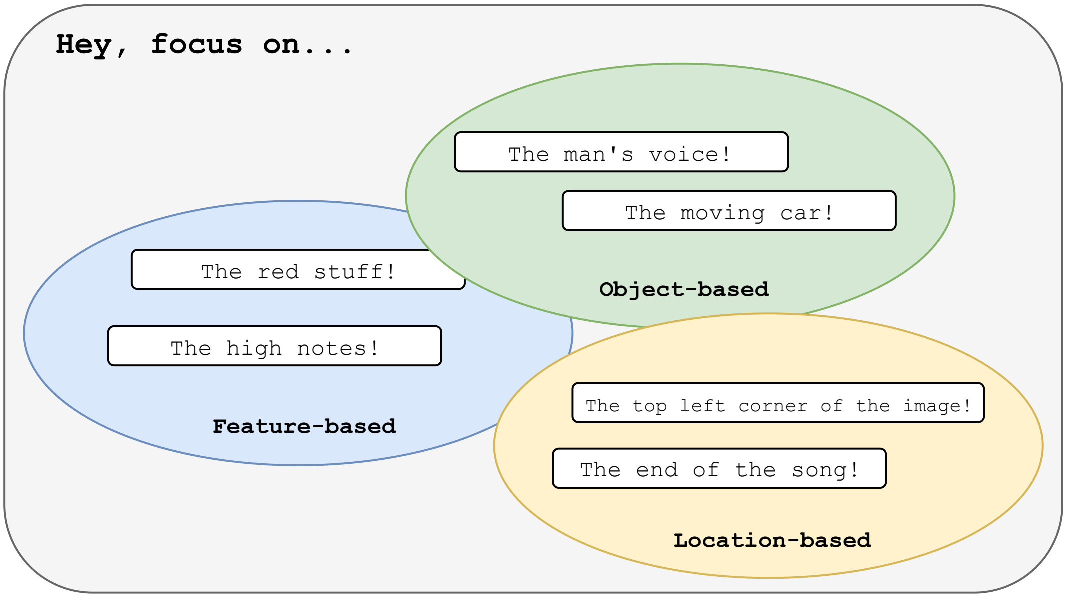

Figure 4: Examples of data selection semantic categories.

3.2.2 Selection course along time

The course of selection *along time* in a process can be classified as *ephemeral* or *enduring*.

Ephemeral refers to selection of subjects in the context of a *short time window*. The evolution of selection along time is not important — only the instantaneous selection is. For example, in a task of visual control of a car, an attention process to identify the abrupt appearance of moving obstacles could be considered ephemeral since it does not take into account much context and only happens during a short time window.

Enduring refers to the process of selection of objects over a *long time window*. The enduring focus can be further classified as:

- **Oriented**: An arbitrary focus sequence so as to complete a certain task.
- **Sustained**: Focus restrained to a subset of the targets.
- **Divided**: Focus alternating among a subset of the targets.

Using the example of visual control of a car, constantly focusing on the road would be considered *sustained* selection. Reading the signs of the road is an *oriented* process since the focus must correctly follow the sequence of letters. Alternatting focus among the mirrors, the speedometer and the road can be considered to be *divided* selection.

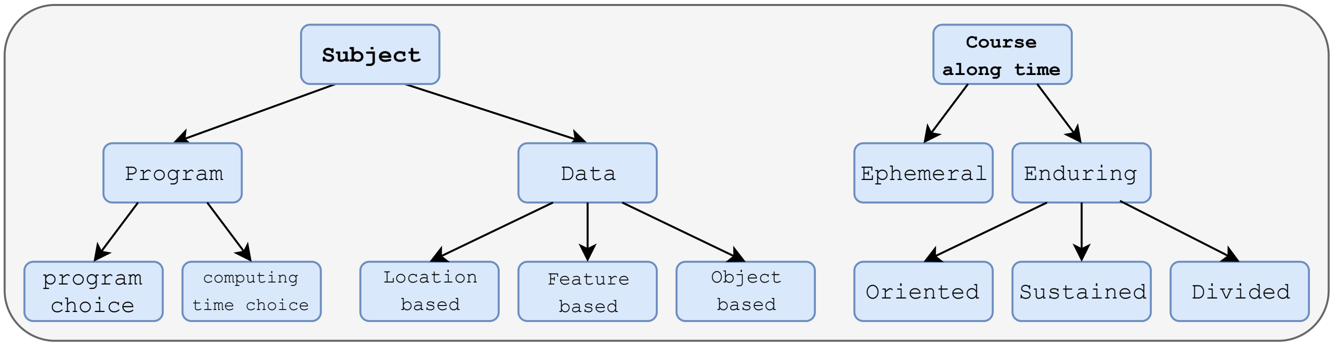

Figure 5: Taxonomy of selection semantics.

3.3 Modules and semantics: examples

Program choice — Selection of a single program: In this example, the system executes only one program at once and must select the next task to be executed. We can consider the target set T to be the set of possible N programs and use hard attention: $A \in \{0, 1\}$, with $0 \leq \sum_{i=1}^{N} \alpha_{ti} \leq 1$.

Program choice — allocation of computing time: In this example, the system contains only one program p to be executed but must decide how much computation to "spend" (from the "budget" B) on the program at a certain timestep. We can set the target set T to be $\{p, nop\}$ (considering *nop* to be doing nothing) and use soft attention: $A \in [0, 1]$, with $\alpha_{t1} + \alpha_{t2} = 1$. Thus, the amount of computation can be calculated as $\alpha_{t1}B$.

Selection of data — focusing on part of an image: In this example, the system must select a window of size $H \times W$ of the image to further perform classification. The focus targets τ_t at time t may represent the set of pixels p_t of the current input image with $N \geq H \times W$ pixels. We can use hard attention and set the focus output set $A \in \{0, 1\}$, with $\sum_{i=1}^{N} \alpha_{ti} = 1$, so that $\alpha_{ti} = 1$ if and only if the window should be centered at pixel p_{ti}.

Feature-based selection of data — focusing more on certain colors: In this example, the system must give weights (summing to 1) to the channels R, G, B of the input image. The focus targets τ_t at time t may represent the respective channels: $\{R_t, G_t, B_t\}$ use soft attention: $A \in [0, 1]$, with $\alpha_{t1} + \alpha_{t2} + \alpha_{t3} = 1$.

Ephemeral selection: Such type of selection takes place during a short time window and state is usually not important. Thus, we can often simply model it by means of a module with *stateless* selection.

Enduring selection: Selection flows which take place along a long time window usually require information about the past, so the module implementing it must often provide *stateful* selection.

Sustained selection: Sustained flow can be achieved, for example, by using one attentional module followed by another. Suppose there are M target elements to be selected and we want to restrict our analysis to a subset of $N \leq M$ elements. The fist module selects such subset in a fixed manner for some amount of time: $A \in \{0, 1\}$, with $\sum_{i=1}^{M} \alpha_{ti} = N$. The target set of the next module is then the set of selected elements: $\{\tau_{ti} : \alpha_{ti} = 1\}$.

3.3.1 Example of an entire system

Figure 3.3.1 shows the diagram of a possible system with attention. The module *TaskATT* uses hard selection to choose 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*, which uses soft selectin 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. Semantically speaking, the selections executed by the modules provide *program choice* and *data selection* capabilities.

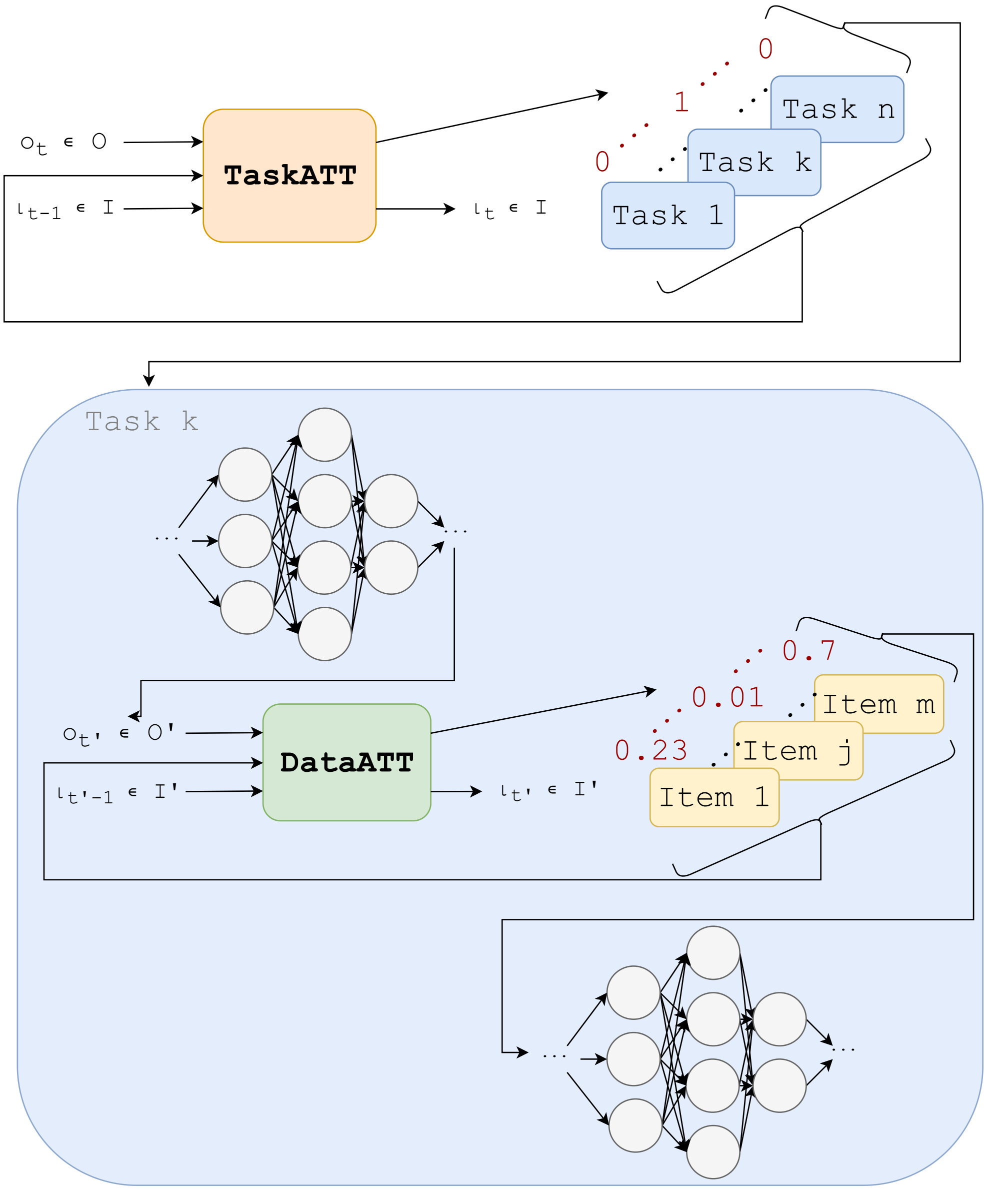

Figure 6: Example of a system that uses attention.

4 Validating the framework

In this section, we investigate some recent works on attention and see how they fit in our framework.

4.1 Image Caption Generation

The work [4] 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 in order to produce another vector to be used in further computations.

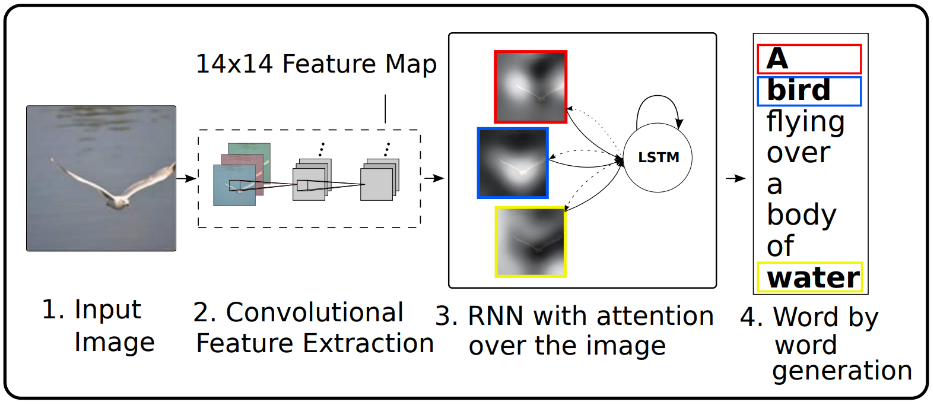
14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

Figure 7: Diagram of natural language image description using Attention (from [1]).

While there are many abstract processing substeps in the process, the *end-to-end* effect is that of a selection with *orienting* course along time, targetting *data* on an *object-based* manner. Figure 4.1 illustrates the proposed model.

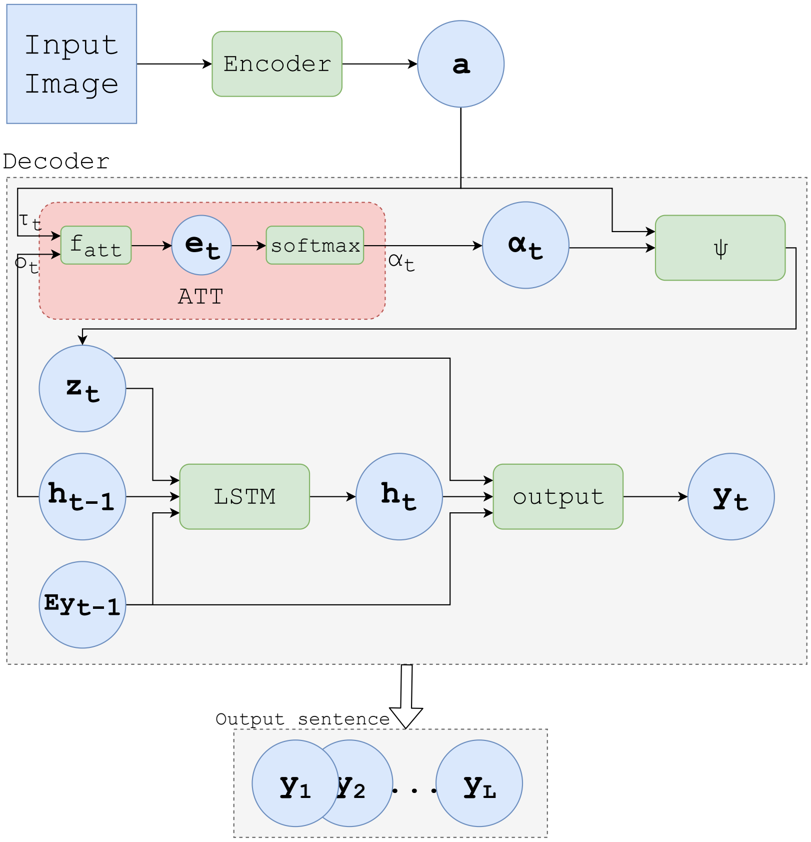

Figure 8: Model proposed for image captioning in [4] with attentional module.

Figure 4.1 illustrates the model proposed in the 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.

4.2 Adaptive Computation Time

The work [2] 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 "spent" 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 $p_{t',t}$.

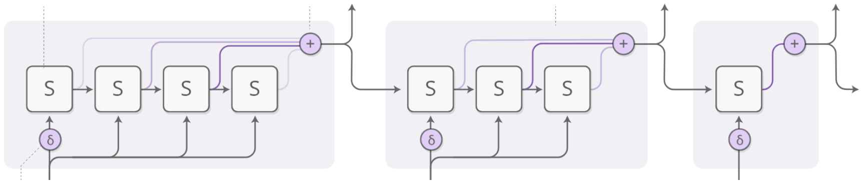

Figure 9: Adaptive computation time process illustration.

The attention component provides two types of selection: *computing time choice oriented* over time and *ephemeral* data selection.

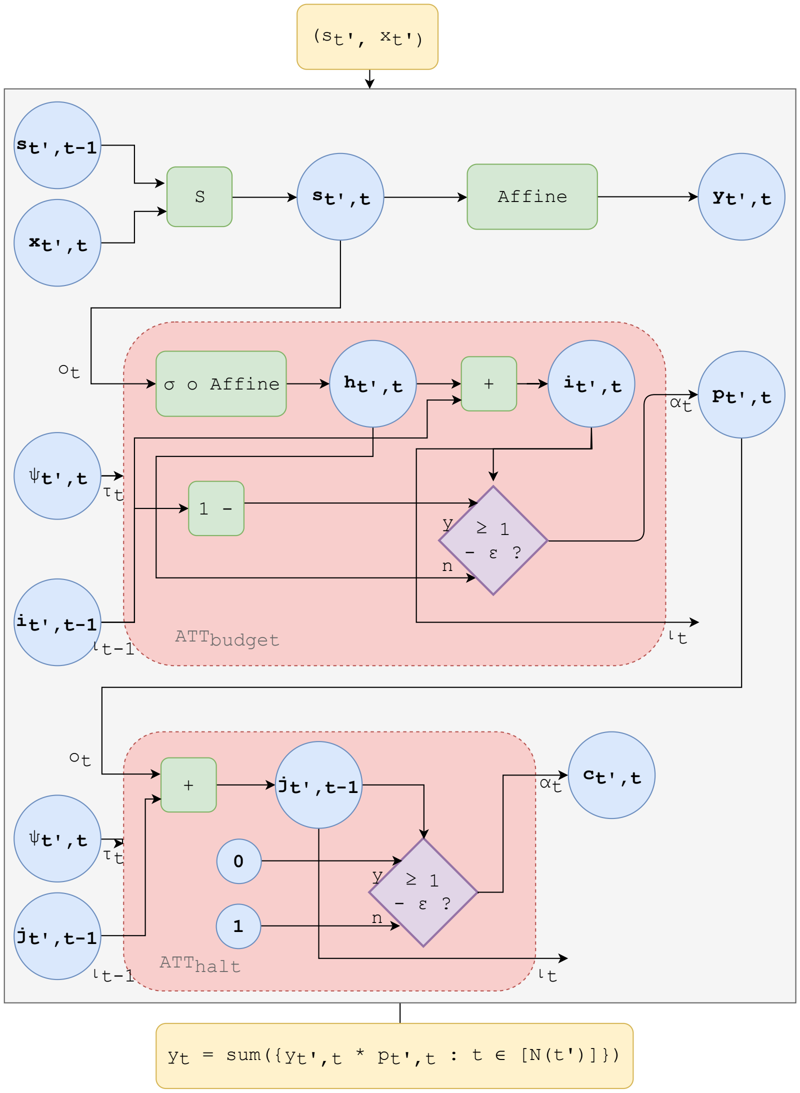

Figure 10: Model proposed for image captioning in [2] with attentional module.

Figure 4.2 illustrates the model proposed in the work. The proposed model can be thought 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 — an 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*.

4.3 Recurrent Attention Model of Visual Attention

The work [3] 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.

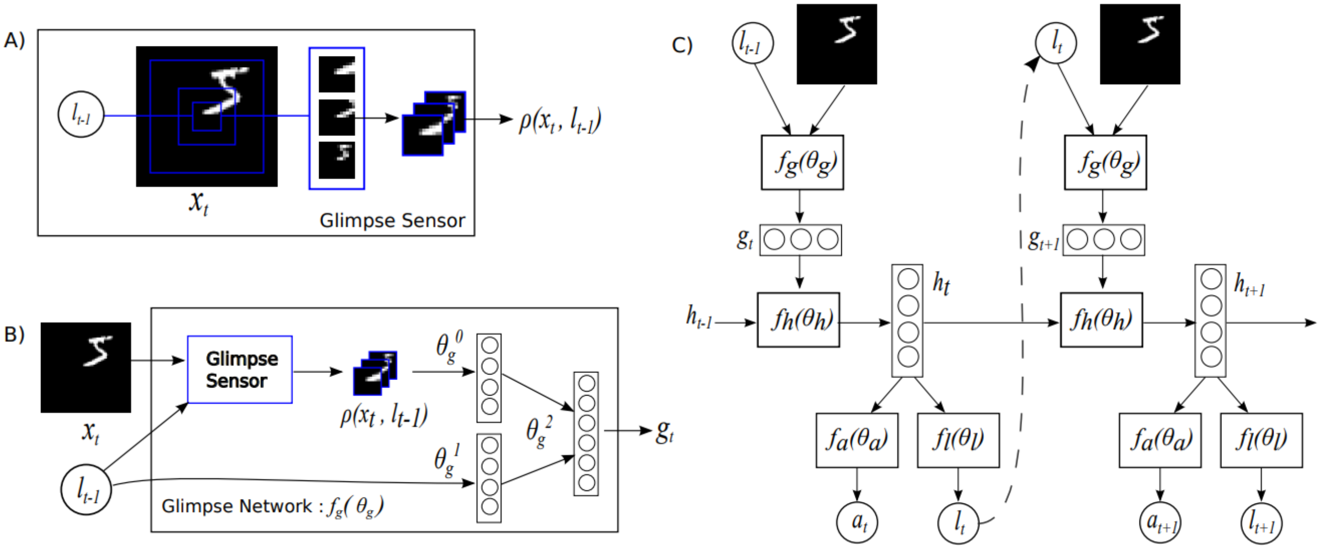

Figure 11: General recurrent architecture proposed in [3].

The attention component in the proposed model can be thought as providing *oriented* selection of *data* in a *location-based* manner.

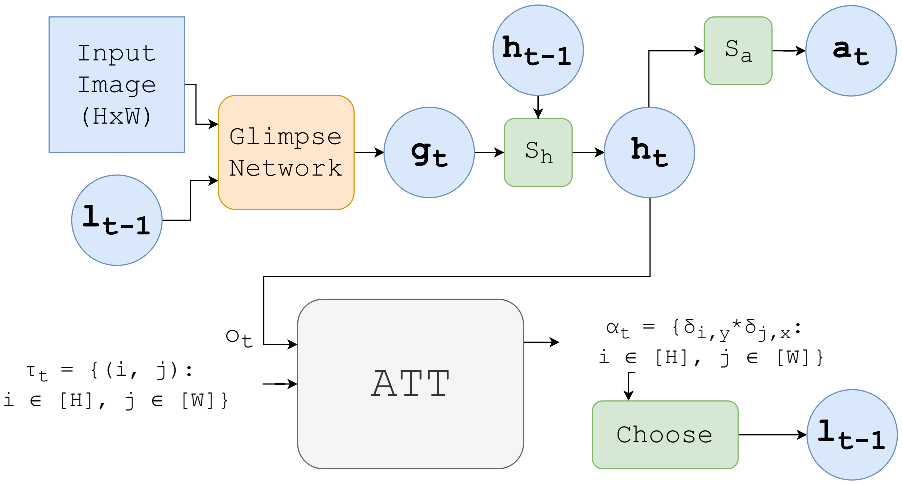

Figure 12: General recurrent architecture proposed in [3] with attentional module.

Figure 4.3 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.

Tables 1 and 2 summarize the categorizations of the works cited above.

Table 1: Selection taxonomy of cited works.

Work	Continuity	State profile
[4]	Soft	Stateful
[2]	Hard	Stateful
[2]	Soft	Stateful
[3]	Hard	Stateful

Table 2: Selection semantics taxonomy of cited works.

Work	Course along time	Semantic target
[4]	Enguring/Oriented	Data/Object-based
[2]	Enduring/Oriented	Program/Computing time choice
[2]	Ephemeral	Data
[3]	Enguring/Oriented	Data/Location-based

References

[1] 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`.

[2] 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).

[3] 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).

[4] 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`.