Location Attention for Extrapolation to Longer Sequences

Yann Dubois
University of Cambridge
yanndubois96@gmail.com

Gautier Dagan
University of Amsterdam
gautier.dagan@gmail.com

Dieuwke Hupkes[*]
ILLC, University of Amsterdam
d.hupkes@uva.nl

Elia Bruni[*]
Universitat Pompeu Fabra
elia.bruni@gmail.com

Abstract

Neural networks are surprisingly good at interpolating and perform remarkably well when the training set examples resemble those in the test set. However, they are often unable to extrapolate patterns beyond the seen data, even when the abstractions required for such patterns are simple. In this paper, we first review the notion of extrapolation, why it is important, and how one could hope to tackle it. We then focus on a specific type of extrapolation, which is especially useful for natural language processing: generalization to sequences longer than those seen during training. We hypothesize that models with a separate content- and location-based attention are more likely to extrapolate than those with common attention mechanisms. We empirically support our claim for recurrent *seq2seq* models with our proposed attention on variants of the *Lookup Table* task. This sheds light on some striking failures of neural models for sequences and on possible methods to approaching such issues.

1 Introduction

It is indisputable that, in recent years, neural network research has made stunning progress on a wide variety of tasks that require to process sequential inputs, such as machine translation (Sutskever et al., 2014) and speech recognition (Graves et al., 2013). However, many researchers have questioned the forms of generalization that neural networks exhibit, which significantly diverges from human-like generalization (Lake and Baroni, 2017; Geirhos et al., 2018). This discrepancy with human-like generalization is particularly true when it comes to extrapolating "outside" the training space (DeLosh et al., 1997; Marcus, 1998).

As neural networks are powerful memorizers (Zhang et al., 2017) and easily learn superficial statistical cues (Jo and Bengio, 2017), testing extrapolation and generalization to samples from the long tails of a distribution might be the only way of quantifying their capacity of abstract reasoning (Santoro et al., 2018).

Despite this benefit, little work has been done in extrapolation. A possible explanation is that the probability of encountering a test example in the extrapolation setting seems low when the training set $\mathcal{D}$ is large.[1] However, such an argument fails to consider the high cost of error in extrapolation settings, and this can be a barrier for real-world scenarios (e.g., self-driving cars).

In this paper, we focus on extrapolation in sequences. More precisely, how to generalize sequence-to-sequence predictors to inputs of length $n_* > n_\mathcal{D}$, where $n_\mathcal{D}$ denotes the length of the longest sequence in the training set. Such extrapolation is crucial for language acquisition, where humans have limited learning resources to account for the unbounded nature of language. To successfully generalize, a language learner needs to process new and potentially longer sentences than previously encountered ones (Chomsky, 1956).

Accounting for this unbounded nature of language is challenging for neural networks. This issue has recently been uncovered for seq2seq models by looking at simple artificial tasks (Lake and Baroni, 2018; Liska et al., 2018; Weber et al., 2018). Liska et al. (2018) find that seq2seq architectures can converge to local minima that generalize, but rarely do. This suggests that neural networks could generalize but lack inductive biases that favor extrapolatable behavior.

In the following sections, we review the concepts of attention and extrapolation. We then argue that

[*] Shared senior authorship

[1] Extrapolation is still prevalent in practical scenarios as high-dimensional problems would typically require an exponentially large $\mathcal{D}$ to be representative, and the underlying distribution may vary over time (Hooker, 2004).

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 403–413
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

current attention mechanisms, which are mainly responsible for recent successes in natural language processing (NLP), are unlikely to extrapolate as they depend on the *content* of trained embeddings. This leads us to introduce a novel *location*-based attention that is loosely inspired by human visual attention. To avoid gaining extrapolation capabilities at the cost of expressivity, we introduce an *attention mixer* that combines content- and position-based attentions. Finally, we show that recurrent models equipped with this new attention mechanism can extrapolate to longer sequences.

2 Extrapolation

Extrapolation is often used but rarely formally defined. Ebert et al. (2014) have found that when extrapolation is explicitly defined, it often refers to points outside a hull delimited by the training set. E.g., rectangular hull, concave hull, or convex hull. In this work we use the rectangle hull definition (Brooks et al., 1988), as any model which is extrapolatable for this region would also be extrapolatable for the convex and concave definition.

Given any finite training dataset $\mathcal{D} := \{\mathbf{x}^{(n)}\}_{n=1}^{N} \subset \mathbb{R}^d$, we define the interpolation domain to be the d-dimensional interval $\mathcal{I}_{inter} := \prod_{i=1}^{d}[\min_n x_i^{(n)}, \max_n x_i^{(n)}]$ and the **extrapolation domain** its complement $\mathcal{I}_{extra} := \mathbb{R}^d \setminus \mathcal{I}_{inter}$. In other words, we define a test example $\mathbf{x}^*$ to be in the extrapolation setting if *at least* one of its features x_j^* is larger or smaller than any values it took during training (Figure 1).

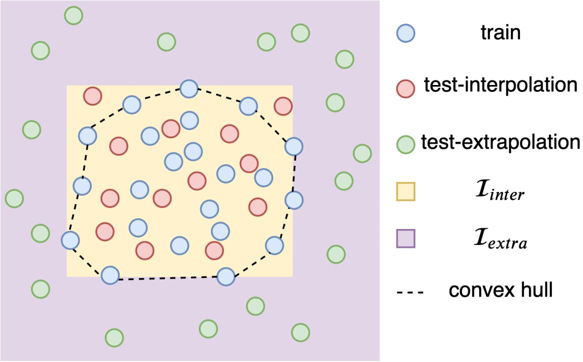


Figure 1: Schematic extrapolation setting for $d = 2$.

Throughout this paper, we assume that neural networks with inputs or temporary representations in $\mathcal{I}_{extra}$ will break. Indeed, for a given target function $t : \mathbb{R}^d \to \mathbb{R}$ to approximate, there is an infinite amount of predictors that satisfy $f(\mathbf{x}) = t(\mathbf{x}),\ \forall \mathbf{x} \in \mathcal{I}_{inter} \subset \mathbb{R}^d$. Without any additional constraints, it is thus extremely unlikely that $f(\mathbf{x}) = t(\mathbf{x}),\ \forall \mathbf{x} \in \mathbb{R}^d$. This could explain why neural networks have empirically been found to break in extrapolation settings (Lohninger, 1999; Hettiarachchi et al., 2005; Mitchell et al., 2018).

The rest of the paper discusses how to constrain representations used by our neural models [2] to be in $\mathcal{I}_{inter}$ regardless of the source sentence length, without decreasing their expressivity.

3 Desiderata

First and foremost, we would like a model that can extrapolate to sequences longer than the longest training one $n_{\mathcal{D}}$ (*Extrapolation Constraint*). As previously discussed, models with inputs or temporary representations in $\mathcal{I}_{extra}$ will very likely break. To satisfy the extrapolation constraint, neural models should thus not depend on features that take values in $\mathcal{I}_{extra}$ for sequences longer than $n_{\mathcal{D}}$.

Second, our model should be able to learn very complex positional attention patterns (*Positional Patterns Constraint*). Finally, although the position of words in a sentence is important, many tasks depend on their semantics. The model should thus still be able to learn content-based attention patterns (*Content Patterns Constraint*).

In the following section, we review previously proposed attention-mechanism and discuss why they do not fulfill the three aforementioned desired properties.

4 Attention Mechanisms

An attention mechanism (or attender) takes as input a matrix of keys $\mathrm{K} := \{\mathbf{k}_s^T\}_{s=1}^{n_s} \in \mathbb{R}^{n_s \times d}$ and a query $\mathbf{q}_t \in \mathbb{R}^d$, and outputs a probability mass function $\boldsymbol{\alpha}_t \in \mathbb{R}^{n_s}$ that will weight a set of values $\mathrm{V} := \{\mathbf{v}_s^T\}_{s=1}^{n_s} \in \mathbb{R}^{n_s \times d_v}$ to generate a glimpse vector $\mathbf{g}_t \in \mathbb{R}^{d_v}$ used for downstream tasks. Following Graves et al. (2014), it is useful to think of the attender as a memory access module, $\boldsymbol{\alpha}_t$ as the soft address and $\mathbf{g}_t$ as the accessed vector.

$$\mathbf{g}_t := \sum_{s=1}^{n_s} \mathbf{v}_s \text{attender}(\mathbf{k}_s, \mathbf{q}_t) = \mathrm{V}\boldsymbol{\alpha}_t \qquad (1)$$

Figure 2 illustrates attention in a recurrent seq2seq (Cho et al., 2014), which we will use for our experiments. Both the keys and the values correspond to the set of encoder hidden states

[2] Although the sentence length is a scalar, the temporary representations (outputs of a hidden layer) are high dimensional.

$\mathrm{K} = \mathrm{V} = \mathrm{E} := \{\mathbf{e}_s^T\}_{s=1}^{n_s}$, while the query corresponds to the current decoder hidden state $\mathbf{q}_t = \mathbf{d}_t$.

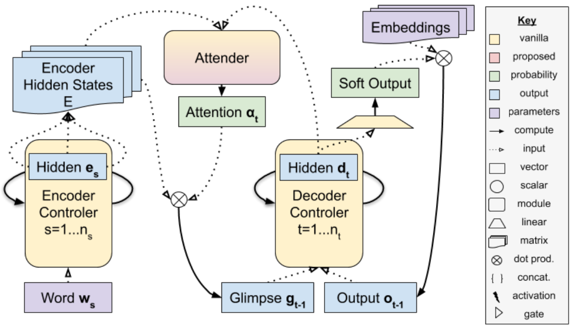


Figure 2: Attender in a recurrent seq2seq.

4.1 Content Attention

Most attention mechanisms compute "content-based addressing" (associative memory) that depend on (partial) matches of the key and query. They take as input K and $\mathbf{q}_t$ and output a semantic-based attention $\boldsymbol{\gamma}_t \in \mathbb{R}^{n_s}$. For example, if you wanted to translate a scientific paper, you could understand the main point of the text without remembering the specific technical terms that were used. When translating, you would go back to the text and translate the jargon by knowing what to look for.

A number of content-based have been proposed, they usually differ in a score that quantifies the match between $\mathbf{k}_s, \mathbf{q}_t$ through multinomial logits:

$$\boldsymbol{\gamma}_t := \{\mathrm{softmax}(\mathrm{score}(\mathbf{k}_s, \mathbf{q}_t))\}_{s=0}^{n_s-1} \tag{2}$$

$$\mathrm{score}(\mathbf{k}_s, \mathbf{q}_t) :=$$

$$\begin{cases} \mathbf{u}^T \tanh([\widetilde{\mathbf{k}}_s; \widetilde{\mathbf{q}}_t]) & \textit{Additive } \text{Bahdanau et al. (2015)} \\ \mathbf{k}_s^T \widetilde{\mathbf{q}}_t & \textit{Multiplicative } \text{Luong et al. (2015)} \\ \frac{\mathbf{k}_s^T \mathbf{q}_t}{\sqrt{d}} & \textit{S. Dot Prod. } \text{Vaswani et al. (2017)} \end{cases} \tag{3}$$

Where $\widetilde{\mathbf{x}}$ is a shorthand for $W\mathbf{x}$.

4.2 Location Attention

A location (or position) attention mechanism computes "location-based addressing" (random access memory) that depend on the index of the key. It takes as input $\mathbf{q}_t$ and outputs a location attention $\boldsymbol{\lambda}_t \in \mathbb{R}^{n_s}$. Intuitively, it decides which value to retrieve based on its index. For example, in German sentences, the verb goes at the end of the sentence, after a subordinate clause. When translating from German to English, it might thus make sense to directly attend to the last word in the German source sentence after encoding a subordinate clause. There are many other cases where attending to words based on their positions seems important. E.g. translating from subject-object-verb to subject-verb-object languages, or understanding the emphasis in some languages.

Despite the importance of word ordering in natural language, location-based attention is not common in *seq2seq* frameworks. This is probably because content-based attention can emulate location-based attention in the usual interpolation setting. Indeed, it can learn to encode a positional embedding in the hidden states of the encoder through some internal "counter". This counter is unlikely to work in the extrapolation regime,[3] we, therefore, investigate other types of location-attention that could satisfy the extrapolation constraint.

Luong et al. (2015) proposed a location-based attention by using Equation 2 with a score that is independent of the key $\mathrm{score}(\mathbf{k}_s, \mathbf{q}_t) = \mathbf{w}^T\mathbf{q}_t$. They restrict themselves to sequences of the same length, which is not of interest to our work. Such a mechanism could be extended to sequences of varying lengths but would still lack extrapolation capacity as the model still has to learn to embed the location of the index it wants to retrieve.

The Neural Turing Machine (Graves et al., 2014), post-processes the content attention by shifting its location by a predicted number of steps. We use a similar mechanism, which is extrapolatable due to the independence of the sequence length. Nevertheless, on its own, it does not allow positional-only patterns in variable-length sentences. For example, it cannot attend to the i^{th} word irrespective of the sentence length. The same argument holds for other location-based attention developed for architectures with an external memory (Sukhbaatar et al., 2015).

More recently, many location-based attention have been proposed in self-attention mechanism. These methods are usually based on sinusoidal encodings (SE), which have been proposed to take into account the word positions while bypassing the need for recurrences in encoder-decoder frameworks. In this paper, we will consider the *transformer* and *transformerXL* (relative SE) attention,

[3] This assumption can depend on the architecture and the inductive bias it provides (Weiss et al., 2018). For our task, we found that the assumption held for both LSTM and GRU.

which are computed as follows.

$$\text{score}(\mathbf{k}_s, \mathbf{q}_t) :=$$

$$\begin{cases} \frac{(\mathbf{k}_s+\mathbf{p}_s)^T(\mathbf{q}_t+\mathbf{p}_t)}{\sqrt{d}} & \textit{Transformer} \text{ (Vaswani et al., 2017)} \\ \frac{(\widetilde{\mathbf{k}}_s+\widetilde{\mathbf{p}}_{s-t})^T(\widetilde{\mathbf{q}}_t+b)}{\sqrt{d}} & \textit{TransformerXL} \text{ (Dai et al., 2019)} \end{cases} \tag{4}$$

Where $\mathbf{p}_t$ is a positional encoding with sinusoidals of different frequencies at every dimension. Although powerful, the sinusoidal encoding and its variants (Shaw et al., 2018; Dai et al., 2019) lack the ability to model location patterns that depend on general word position such as "look at the i^{th} word (after ...)" in the extrapolation setting. Indeed, the sinusoidal encoding for any fixed offset $\mathbf{p}_{t+k}$ is linear in $\mathbf{p}_t$ but not in k.

Location-based processing of attention has also been proposed as a way of constraining content-based attention to some (soft) window. Yang et al. (2018) achieve it by multiplying the content attention by the weights of a predicted Gaussian such that the model has an inductive bias towards attending to words that are close to each other. Sukhbaatar et al. (2019) use a piece-wise window to decrease the computational complexity of the model. These methods nevertheless solve a fundamentally different problem and do not allow location-only extrapolatable patterns of attention.

5 Model

In this section, we propose a location attender that can satisfy the extrapolation and positional patterns constraint. We then discuss how to incorporate content attention to satisfy the content patterns constraint.

5.1 Location Attender

We would like our position attention to be loosely reminiscent of human attention, whereby we sequentially focus on a single area of the input (e.g., words or pixels) but vaguely perceive neighboring inputs due to the eccentricity effect (Carrasco et al., 1995). The visual acuity of humans is uni-modal, symmetric, and spikes at the fovea, which corresponds to a 0° retinal eccentricity. We model this visual acuity using a Gaussian Probability Density Function (PDF) similarly to Mnih et al. (2014).[4] I.e. for each step, the Location Attender models a Gaussian attention over the relative word positions.

Specifically, it generates a mean μ_t and standard deviation σ_t, which are used to compute the location attention given the values of the PDFs at the relative indices $r_s := \frac{s}{n_s-1}$ of the keys:

$$\boldsymbol{\lambda}_t := \{\text{PDF}_{\mu_t,\sigma_t}(r_s)\}_{s=0}^{n_s-1}$$

Using relative indices r_s instead of the absolute ones s is crucial such that the generated μ_t is bounded (in $[0, 1]$), thereby satisfying the extrapolation constraint.

This model, unfortunately, fails to satisfy the positional patterns constraint, as it only allows patterns of attention based on percentile positions. E.g., it can decide to attend to the 10%-percentile word but not to the 2^{nd} word. This incapacity to satisfy the position pattern constraint is a general issue with commonly used attention mechanisms (including sinusoidal-based) that only becomes apparent when dealing with complex extrapolation patterns.

To have a general attention mechanism, we need a μ_t that can: i) attend to locations based on absolute positions; ii) attend to locations based on percentile positions; iii) attend to positions based on the previous attention. We achieve this by defining one building block for each of those requirements ($\mathbf{b}_t$) such that their weighted average forms μ_t, and the weights $\boldsymbol{\rho}_t$ are bounded outputs of the model. The three building blocks are:

- The step size $\frac{1}{n_s-1}$ between words allows the attention mechanism to depend on absolute positions. The generated weight is an integer, which dictates the additional number of steps to take.

- The bias term 1 enables the model to use percentile positions. The generated weight gates it (on or off).

- The average position of the previous attention $\bar{\boldsymbol{\alpha}}_{t-1}$ that is gated by the generated weight. This ensures that the model can attend using absolute positions to words at indices not seen during training. E.g., attending to index $n_{\mathcal{D}} + 5$ by first attending to $n_{\mathcal{D}}$ then $\bar{\boldsymbol{\alpha}}_{t-1} + 5$.

The weights $\boldsymbol{\rho}_t$ are generated using a Gated Recurrent Unit (GRU) (Cho et al., 2014). μ_t is clamped to $[0, 1]$ by a linear function to yield interpretable and extrapolatable behaviour. We also

[4] Visual acuity is distributed in a Laplace-like distribution, but initial experiments were more encouraging using a Gaussian.

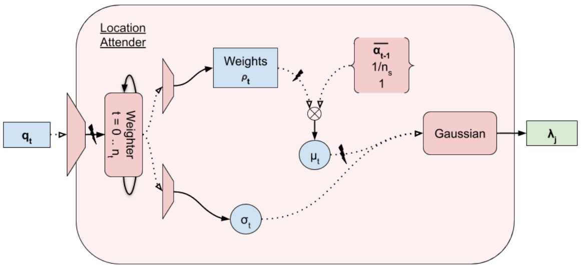


Figure 3: Proposed Location Attender. Given a resized query, the Weighter outputs the standard deviation σ_t and $\boldsymbol{\rho}_t$ which will weight the building blocks $\mathbf{b}_t$ to compute the mean μ_t. μ_t and σ_t parametrize a Gaussian PDF used to compute the location attention $\boldsymbol{\lambda}_t$.

force $\sigma_t > min_\sigma$ and normalize it by n_s which respectively avoids division by 0 and makes σ_t comparable regardless of n_s. A graphical overview of the Location Attender can be seen in Figure 3. Formally:

$$\begin{aligned}
\boldsymbol{\omega}_t &:= \text{GRU}\left(\text{ReLU}\left(\text{W}^{(resize)}\mathbf{q}_t\right)\right) \\
\sigma_t &:= \frac{\text{ReLU}(\text{W}^{(\sigma)}\boldsymbol{\omega}_t) + min_\sigma}{n_s} \\
\boldsymbol{\rho}_t &:= a(\text{W}^{(\rho)}\boldsymbol{\omega}_t) \\
\mathbf{b}_t &:= \{\bar{\boldsymbol{\alpha}}_{t-1}; \frac{1}{n_s - 1}; 1\} \\
\mu_t &:= \text{clamp}(\boldsymbol{\rho}_t^T \mathbf{b}_t) \\
\lambda_{st} &:= \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left(\frac{-(\frac{s}{n_s-1} - \mu_t)^2}{2\sigma_t^2}\right)
\end{aligned}$$

Where *clamp* is a leaky clamping (2 leaky ReLUs) and $\text{min}_\sigma = 0.27$. a is the activation function that forces each of the three dimensions of $\boldsymbol{\rho}_t$ to take on the desired values. Namely a sigmoid activation for the gates, and the following "soft-staircase" [5] to force the weights of the step size to be approximately integers (Figure 4):

$$softstair(x) := \lfloor x \rfloor + \text{sigmoid}(20(x - 0.5 - \lfloor x \rfloor))$$

5.2 Mix Attender

We enforce the content patterns constraint, by using a convex combination of content and location attention (Figure 5):

$$\begin{aligned}
\boldsymbol{\alpha}_t &:= \%_t^{(\lambda)}\boldsymbol{\lambda}_t + (1 - \%_t^{(\lambda)})\boldsymbol{\gamma}_t \\
\%_t^{(\lambda)} &:= \text{sigmoid}(\text{W}^{(\%)}\mathbf{q}_t)
\end{aligned}$$

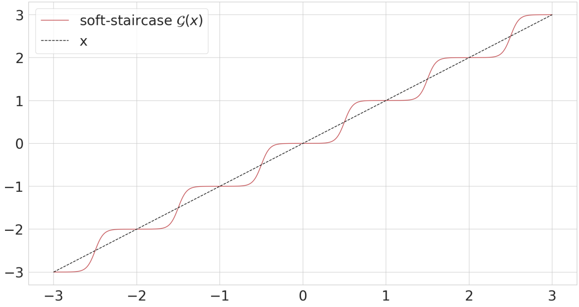


Figure 4: Soft staircase activation function.

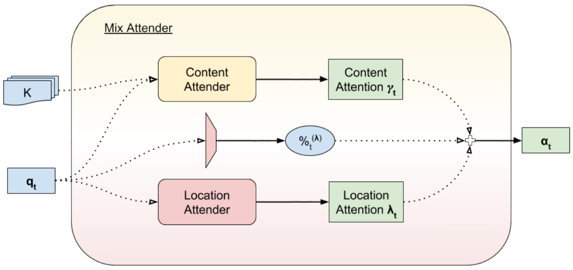


Figure 5: Mix Attender. The output $\boldsymbol{\alpha}_t$ is a convex combination of the content and location attention.

6 Experiments

6.1 Datasets

The fact that humans generate and understand unbounded sentences with a finite experience is often used as proof of the *principle of compositionality* (Szab, 2017). Following this argument, methods that can extrapolate to longer sequences should exhibit some compositionality.

Based on this observation, we evaluate on a compositionality-specific artificial task, *lookup tables* (Liska et al., 2018), but extend it to better quantify extrapolation. [6] This task is especially interesting to us, as there is a clear notion of what a good attention pattern should look like, making it easy to qualitatively and quantitatively analyze attentive models. It is a well-controlled task, which allows us to uncover challenges that prevent models from extrapolating on real-world data.

6.1.1 Long Lookup Tables

The *lookup tables* task consists in sequentially applying k pre-defined lookup table functions. The lookup tables are bijective mappings on the set of

[5] Straight-through estimators (Bengio et al., 2013) and Gumbel-Softmax (Jang et al., 2017; Maddison et al., 2017) performed slightly worst and required predefining the maximum number of steps.

[6] The extended datasets as well as scripts to generate them can be found at `https://github.com/i-machine-think/machine-tasks/tree/master/LongLookupTables`

Input	Target	Target Attention
000 t1 .	000 110 <eos>	0 1 2
110 t1 .	110 110 <eos>	0 1 2
110 t2 .	110 100 <eos>	0 1 2
000 t1 t1 t2 .	000 110 110 100 <eos>	0 1 2 3 4

Table 1: Long lookup table examples.

all 3-bit strings $t_i : \{0,1\}^3 \rightarrow \{0,1\}^3$. For example, if $t_1(000) = 110$ and $t_2(110) = 100$ then $t_2(t_1(000)) = t_2(110) = 100$. Following Hupkes et al. (2018), we write the operations from left to right, as well as add the inputs and temporary steps to the targets. E.g. the previous example corresponds to the input `000 t1 t2` and the target `000 110 100`.

General extrapolatable seq2seq models should be able to terminate by outputting an end of sentence token `<eos>`. We thus append `<eos>` to the targets and a full stop `.` to the inputs. [7]

At each decoding step, the target only depends on the previous output and the current lookup table. E.g. the last decoding step of `000 t1 t2`, only depends on the previous output $110 = t_1(000)$ and the current table t_2. The network thus has to learn the lookup table mappings and use the correct one at each step. The gold standard attention, therefore, corresponds to the position of the current lookup table. Table 1 illustrates a longer example and its correct attention.

The various train and test sets are generated by composing 6 random lookup tables $t_1, \ldots, t_6$ that have as input and output one of the $2^3 = 8$ possible 3-bit strings. Specifically, we use $k = 1 \ldots 4$ composed tables in the training set, $k = 2 \ldots 4$ for the interpolation test sets, and $k = 5 \ldots 9$ for the extrapolation test sets.

There are 5 different extrapolation test sets, depending on their additional lengths compared to the maximum training examples (`long 1`, ..., `long 5`). We randomly select only 5000 possible examples for each of these test sets.

For the interpolation test sets, we select 3000 examples from all possible input-output pairs.

The training set contains all other possible input-output pairs, approximately 10000 examples.

6.1.2 Reversed Lookup Tables

To test whether the attention can generate more complex patterns (investigating the Positional Patterns Constraint), we also introduce a dataset which reverses the order of the inputs in the previous dataset. E.g. the last example in Table 1, would be written as `t2 t1 t1 000 .`, the target would not change, and the attention pattern should be `3 2 1 0 4` (attend to `.` when outputting `<eos>`). Although the change seems minor, we hypothesize that such a setting will be much more complicated as the attention pattern is not monotonic and does not follow the encoding nor the decoding steps. Indeed, in the previous task, the model only needs to learn to match the i^{th} decoding step with the i^{th} encoding step.

6.1.3 Lookup Tables with Noisy Start

Input	Target	Target Attention
000 t2 ! t1 .	000 110 <eos>	0 2 3 4
110 t5 t3 t1 ! t1 .	110 110 <eos>	0 4 5 6
110 ! t2 .	110 100 <eos>	0 2 3
000 t6 t3 ! t1 t1 t2 .	000 110 110 100 <eos>	0 3 4 5 6 7

Table 2: Lookup tables with noisy start examples

Finally, we introduce another variant that also requires content attention (investigating the Content Patterns Constraint). To do so, we augment each training example with a start token "`!`" between the input and the tables in the source sequence. We then add $m \sim \mathcal{U}\{0, 10\}$ tables t_i before the start token. The target outputs were not modified and are thus independent of the added tables. Solving this task requires to first attend to the input, then to the token which follows "`!`" (content attention) and finally proceed with incremental location attention. Examples of the training data are given in Table 2.

6.2 Metrics

The main metric is *sequence accuracy* (*seqAcc*), which corresponds to the accuracy of predicting the entire sequence correctly (including its length). To get insights about how the model works, we will also use two other losses.

Sequence Accuracy Before Eos (*seqAccBE*), which only evaluates the accuracy of the subsequence before the model generated a `<eos>`.

Attention Loss (*attnLoss*), which quantifies the quality of the attention pattern before `<eos>`. It is computed as the mean squared error between the predicted and gold standard attention. [8] The attention loss gives an indication of how far the

[7] This makes the task harder than the one in Hupkes et al. (2018), who force termination after the right amount of steps.

[8] The loss is overly simplistic as it is symmetric around $\bar{\boldsymbol{\alpha}}_t$ even though errors in the temporal direction are less serious as the embeddings contain past information.

model is to the ideal attention patterns required to solve the sequence.

6.3 Architecture and Baselines

Concerning baselines, we use three content attention: additive, multiplicative, scaled dot product (Eq.3). We also have two mixed content-location attention baselines: Transformer and TransformerXL (Eq.4).

To focus on the attention mechanisms, our model and the baselines all use a smaller version of the best performing recurrent seq2seq architecture on the *lookup table* task (Hupkes et al., 2018). The model has never been modified during our experimentation and is schematized in Figure 2. The embeddings are of dimension 64, the recurrent network is a GRU (Cho et al., 2014) with a hidden size of 128, 50% dropout (Srivastava et al., 2014) is applied on the encoder-decoder bottleneck, and a residual connection is used between the inputs (embeddings) and outputs of the encoder. Training consists of 50 epochs with the Adam (Kingma and Ba, 2015) optimizer.

7 Results

7.1 Interpolation

For sanity check, we tested all the baselines and our models (with and without attention mix) on the interpolation setting of the three tasks. Our models and the best baseline (transformer attention) achieved 100% sequence accuracy (*seqAcc*).

7.2 Extrapolation Constraint

The major desired property of our model is to be able to extrapolate. We tested the extrapolation capacity of our location attender by evaluating its *seqAcc* on the long lookup table extrapolation test sets. Figure 6 shows the *seqAcc* of the location attender against the strongest baseline (transformer attention).

As hypothesized, the transformer attention has some extrapolation capacity, but our location attender substantially outperforms it in this simple task. Importantly, the loss in performance in the extrapolation setting for the best baseline is abrupt and goes from 100% to 0% by adding only three tokens to the inputs. This suggests that commonly used models are brittle and cannot even extrapolate by a small amount.

Although the previous results are encouraging, we would like to understand what is holding back our model from perfectly extrapolating (Figure 6).

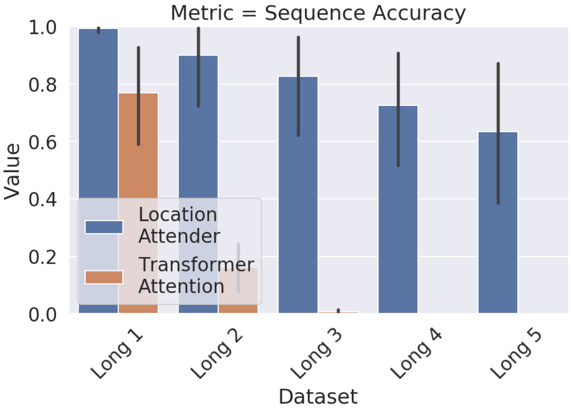


Figure 6: *SeqAcc* for the Location Attender and best baseline on the Long Lookup Tables task (10 runs).

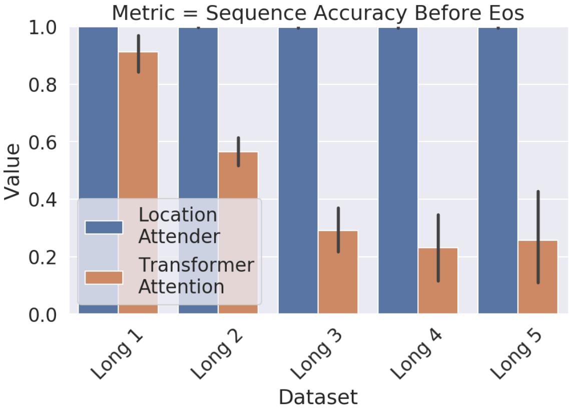


Figure 7: *SeqAccBE* for the Location Attender and best baseline on the Long Lookup Tables task (10 runs).

To do so, we computed the sequence accuracy before `<eos>` (*SeqAccBE*). Figure 7 shows that the model outputs are always correct but that it often terminates decoding too soon, which we will refer to as **the `<eos>` problem**. This suggests that the decoder keeps an internal "counter" to increase the probability of outputting `<eos>` when the decoding step is greater than the ones seen at training time. The model learns this heuristic, which is always correct during training time and can be thought of as a metric hacking. Importantly, it is not a "hard" boundary: the model is often able to extrapolate a couple of steps but usually stops before the correct number of steps.

7.3 Positional and Content Patterns Constraint

Having shown that our model can extrapolate well on a simple task, we would like to investigate whether it can do so for tasks that require more complicated attention patterns such as the reversed and noisy task.

Although the Mix Attender, outperformed all baselines on both tasks, it was not able to get more than 40% and 5% sequence accuracy for `long 1` and `long 2` respectively.

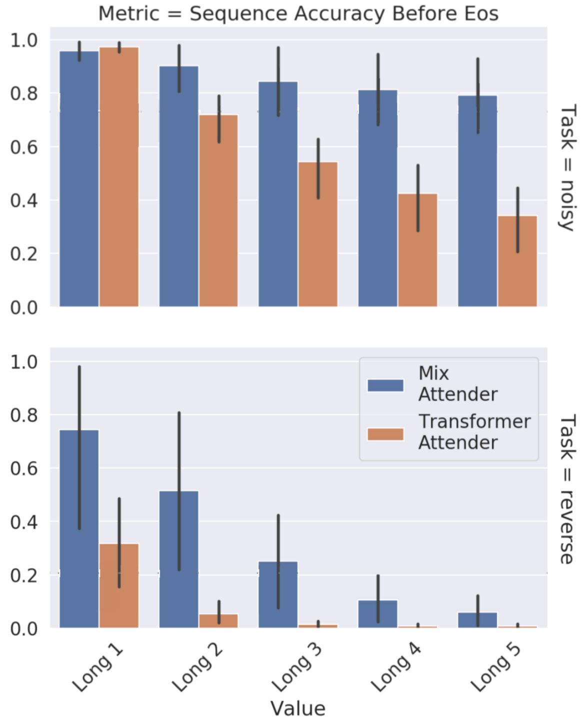


Figure 8: *SeqAccBE* (5 runs) for the Mix Attender and best baseline on the reversed lookup tables (*reverse*) and lookup tables with noisy start (*noisy*).

Figure 8 shows that when considering *seqAccBE*, the Mix Attender is able to extrapolate well in the noisy setting and a little in the reverse setting. This suggests that it is not able to extrapolate well when considering sequence accuracy because it strongly suffers from the `<eos>` problem. This is a recurrent problem in our experiments and is more likely to happen in harder tasks and larger models.

7.4 Attention Pattern

As previously discussed, variants of the lookup table task are especially interesting as we know the gold standard attention pattern. This enables evaluation of attention patterns through the MSE attention loss (*attnLoss*).

Table 3 shows the attention loss averaged over the three tasks. Although not perfect, the Mix Attender performs on average the best across all settings. [9] Crucially, it performs similarly in an interpolation setting and simple extrapolation setting (`long 1`), while all other baselines perform significantly worse after adding a single token. Even in `long 2`, it is competitive with all other attention mechanisms in their interpolation domain. This indicates that the model is indeed able to extrapolate by being more precise with its attention pattern.

Attention	Interp.	Long 1	Long 2	Long 3	Long 4	Long 5
Scaled Dot	5.3	6.3	8.1	10.2	12.6	15.4
Multiplicative	3.1	4.6	6.3	7.9	9.9	12.4
Additive	3.1	8.4	15.6	22.2	28.7	34.8
Transformer	2.8	3.5	6.1	9.1	11.7	13.9
TransformerXL	3.0	3.9	5.3	7.1	9.1	11.4
Mix Attention	**2.1**	**2.2**	**2.9**	**4.1**	**5.3**	**6.7**

Table 3: *AttnLoss* for various attention models averaged over the three datasets and 5 runs.

7.5 Qualitative Analysis

In addition to enabling extrapolation, the temporary variables such as the weight given to each building block are very helpful for debugging the model and improving interpretability.

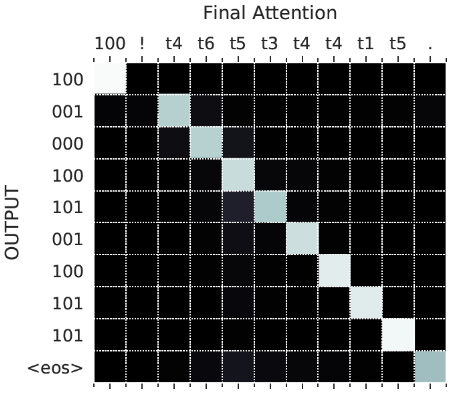
Figure 9 shows the output of a Mix Attender for the *lookup tables with noisy start* task. The input was sampled from the `Long 4` test set. The top-left image shows the final attention. The top-right table shows the value of some interpretable variables at every decoding step. The bottom images correspond to the content and location attention.

The first decoding step uses location attention to attend to the first input. For the next three steps, the model outputs a mixing weight $\%^{(\lambda)} \approx 0$ to focus on content attention. The content attention successfully finds the first non-noisy table (after `!`). [10] It then goes back to using the location attention with $\rho^{(\alpha)} = 1$ and $\rho^{(1/n)} = 1$ to generate a diagonal attention. Finally, it predicts `<eos>` when attending to the end of the input "`.`".

At each step, $\sigma = min_\sigma$ as it does not need to attend to neighboring words for this task. $\%^{(\lambda)}$ is never exactly 0 or 1, such that the model can easily learn to switch between content and location attention as it does not collapse to using a single form of attention.

[9] Some baselines outperformed it in the interpolation settings of specific tasks. Namely, the additive attention in the reversed task and transformer in the noisy task.

[10] A single step of content attention should be sufficient, but the model seems to consistently use three steps.



$\%_t^{(\lambda)}$	μ_t	σ_t	$\rho_t^{(a)}$	$\rho_t^{(\frac{1}{n_s})}$	$\rho_t^{(1)}$
0.98	0.00	0.27	1.07	1.00	0.00
0.01	0.10	0.27	1.08	1.00	0.00
0.00	0.30	0.27	1.08	1.00	0.00
0.02	0.40	0.27	1.09	1.00	0.00
0.72	0.50	0.27	1.09	1.00	0.00
0.83	0.60	0.27	1.09	1.00	0.00
0.91	0.70	0.27	1.09	1.00	0.00
0.91	0.80	0.27	1.09	1.00	0.00
0.96	0.90	0.27	1.09	1.00	0.00
0.71	1.00	0.27	1.09	1.00	0.00

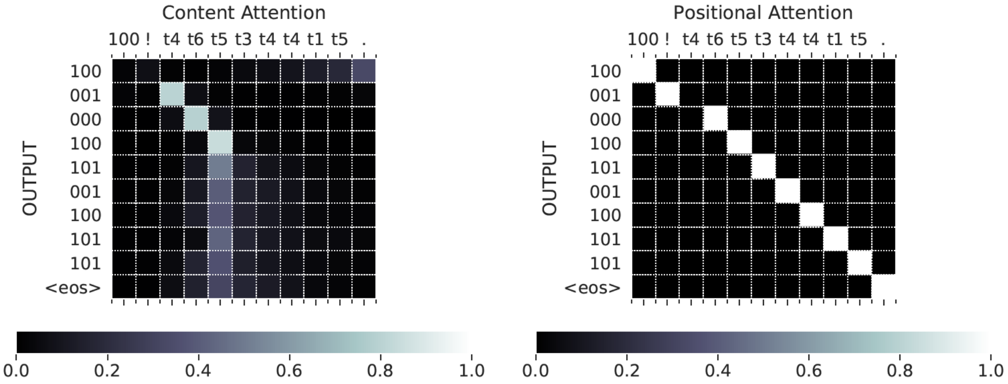


Figure 9: Example output by the attention-mixer for the lookup tables with noisy start task (`Long 4` test set).

8 Discussion

In this paper, we focused on one type of extrapolation, which is especially important in NLP: generalization to longer sequences. We propose a new location-based attention, and show that it can extrapolate better than previous models while learning various attention patterns.

Despite promising initial results, our model is still unable to extrapolate perfectly for harder tasks. By analyzing its behavior, we uncovered an interesting heuristic used by seq2seq models, namely that they keep track of a decoding "counter" to know when to output the <eos> token. This is a bottleneck for extrapolation, suggesting that removing this heuristic is key to reaching perfect extrapolation and should be investigated in future work.

Once the <eos> problem is solved, we could test the model on real-world datasets. It would also be interesting to test such attention mechanisms in self-attentive seq2seq models without recurrence. Finally, as the location attender is not model dependent, it could be pretrained on complex location patterns and incorporated as a plug-and-play module to get extrapolatable position attention.

Taking a step back, we have shown that current deep learning models with common attention mechanisms are unable to extrapolate well on seemingly straightforward tasks. This tends to be overlooked by the field due to standard benchmarks that can be solved using only interpolation. We hope that this paper acts as a reminder that extrapolation is a hard setting that has not been much investigated by the machine learning community. As current methods that memorize and learn superficial cues are unable to extrapolate while humans are, we believe that such a setting might help (and force) the field to come up with more human-like computational models that are capable of abstract reasoning.

Acknowledgments

Dieuwke Hupkes is funded by the Netherlands Organization for Scientific Research (NWO), through a Gravitation Grant 024.001.006 to the Language in Interaction Consortium. Elia Bruni is funded by the European Unions Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 790369 (MAGIC).

References

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, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432.

Daniel G Brooks, Steven S Carroll, and William A Verdini. 1988. Characterizing the domain of a regression model. *The American Statistician*, 42(3):187–190.

Marisa Carrasco, Denise L Evert, Irene Chang, and Svetlana M Katz. 1995. The eccentricity effect: Target eccentricity affects performance on conjunction searches. *Perception & Psychophysics*, 57(8):1241–1261.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL.

Noam Chomsky. 1956. Three models for the description of language. *IRE Trans. Information Theory*, 2(3):113–124.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2978–2988. Association for Computational Linguistics.

Edward L DeLosh, Jerome R Busemeyer, and Mark A McDaniel. 1997. Extrapolation: The sine qua non for abstraction in function learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 23(4):968.

Tobias Ebert, Julian Belz, and Oliver Nelles. 2014. Interpolation and extrapolation: Comparison of definitions and survey of algorithms for convex and concave hulls. In *2014 IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2014, Orlando, FL, USA, December 9-12, 2014*, pages 310–314. IEEE.

Robert Geirhos, Carlos R. Medina Temme, Jonas Rauber, Heiko H. Schütt, Matthias Bethge, and Felix A. Wichmann. 2018. Generalisation in humans and deep neural networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 7549–7561.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. 2013. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, pages 6645–6649. IEEE.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *CoRR*, abs/1410.5401.

P Hettiarachchi, MJ Hall, and AW Minns. 2005. The extrapolation of artificial neural networks for the modelling of rainfallrunoff relationships. *Journal of Hydroinformatics*, 7(4):291–296.

Giles Hooker. 2004. *Diagnostics and extrapolation in machine learning*. stanford university.

Dieuwke Hupkes, Anand Singh, Kris Korrel, Germán Kruszewski, and Elia Bruni. 2018. Learning compositionally through attentive guidance. *CoRR*, abs/1805.09657.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Jason Jo and Yoshua Bengio. 2017. Measuring the tendency of cnns to learn surface statistical regularities. *CoRR*, abs/1711.11561.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Brenden M. Lake and Marco Baroni. 2017. Still not systematic after all these years: On the compositional skills of sequence-to-sequence recurrent networks. *CoRR*, abs/1711.00350.

Brenden M. Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2879–2888. PMLR.

Adam Liska, Germán Kruszewski, and Marco Baroni. 2018. Memorize or generalize? searching for a compositional RNN in a haystack. *CoRR*, abs/1802.06467.

Hans Lohninger. 1999. *Teach/me: Data Analysis*. Springer.

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, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1412–1421. The Association for Computational Linguistics.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Gary F Marcus. 1998. Rethinking eliminative connectionism. *Cognitive psychology*, 37(3):243–282.

Jeff Mitchell, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Extrapolation in NLP. *CoRR*, abs/1805.06648.

Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. 2014. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2204–2212.

Adam Santoro, Felix Hill, David G. T. Barrett, Ari S. Morcos, and Timothy P. Lillicrap. 2018. Measuring abstract reasoning in neural networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4477–4486. PMLR.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 464–468. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. 2019. Adaptive attention span in transformers. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 331–335. Association for Computational Linguistics.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2440–2448.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Zoltn Gendler Szab. 2017. Compositionality. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, summer 2017 edition. Metaphysics Research Lab, Stanford University.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Noah Weber, Leena Shekhar, and Niranjan Balasubramanian. 2018. The fine line between linguistic generalization and failure in Seq2Seq-attention models. In *Proceedings of the Workshop on Generalization in the Age of Deep Learning*, pages 24–27, New Orleans, Louisiana. Association for Computational Linguistics.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2018. On the practical computational power of finite precision rnns for language recognition. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 740–745. Association for Computational Linguistics.

Baosong Yang, Zhaopeng Tu, Derek F. Wong, Fandong Meng, Lidia S. Chao, and Tong Zhang. 2018. Modeling localness for self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4449–4458. Association for Computational Linguistics.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization.