Influence Paths for Characterizing Subject-Verb Number Agreement in LSTM Language Models

Kaiji Lu
Carnegie Mellon University
kaijil@andrew.cmu.edu

Piotr Mardziel
Carnegie Mellon University
piotrm@gmail.com

Klas Leino
Carnegie Mellon University
kleino@cs.cmu.edu

Matt Fredrikson
Carnegie Mellon University
mfredrik@cmu.edu

Anupam Datta
Carnegie Mellon University
danupam@cmu.edu

Abstract

LSTM-based recurrent neural networks are the state-of-the-art for many natural language processing (NLP) tasks. Despite their performance, it is unclear whether, or how, LSTMs learn structural features of natural languages such as subject-verb number agreement in English. Lacking this understanding, the generality of LSTMs on this task and their suitability for related tasks remains uncertain. Further, errors cannot be properly attributed to a lack of structural capability, training data omissions, or other exceptional faults. We introduce *influence paths*, a causal account of structural properties as carried by paths across gates and neurons of a recurrent neural network. The approach refines the notion of influence (the subject's grammatical number has influence on the grammatical number of the subsequent verb) into a set of gate-level or neuron-level paths. The set localizes and segments the concept (e.g., subject-verb agreement), its constituent elements (e.g., the subject), and related or interfering elements (e.g., attractors). We exemplify the methodology on a widely-studied multi-layer LSTM language model, demonstrating its accounting for subject-verb number agreement. The results offer both a finer and a more complete view of an LSTM's handling of this structural aspect of the English language than prior results based on diagnostic classifiers and ablation.

1 Introduction

Traditional rule-based NLP techniques can capture syntactic structures, while statistical NLP techniques, such as n-gram models, can heuristically integrate semantics of a natural language. Modern RNN-based models such as Long Short-Term Memory (LSTM) models are tasked with incorporating both semantic features from the statistical associations in their training corpus, and structural features generalized from the same.

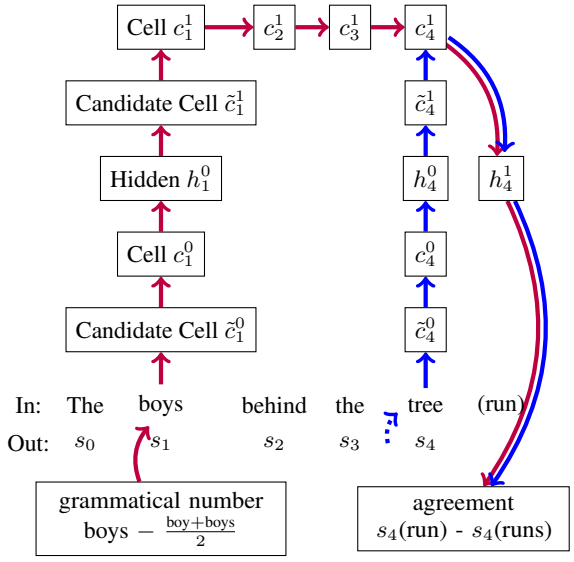

Figure 1: Subject-verb agreement task for a 2-layer LSTM language model, and primary paths across various LSTM gates implementing subject-verb number agreement. A language model assigns score s to each word. Agreement is the score of the correctly numbered verb minus that of the incorrectly numbered verb.

Despite evidence that LSTMs can capture syntactic rules in artificial languages (Gers and Schmidhuber, 2001), it is unclear whether they are as capable in natural languages (Linzen et al., 2016; Lakretz et al., 2019) in the context of rules such as subject-verb number agreement, especially when not supervised for the particular feature. The incongruence derives from this central question: *does an LSTM language model's apparent performance in subject-verb number agreement derive from statistical heuristics (like n-gram models) or from generalized knowledge (like rule-based models)?*

Recent work has begun addressing this question (Linzen et al., 2016) in the context of *language models*: models tasked with modeling the likelihood of the next word following a sequence of words as expected in a natural language (see Figure 1, bottom). *Subject-verb number agreement* dictates that the verb associated with a given subject

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4748–4757

should match its number (e.g., in Figure 1, the verb "run" should match with the subject "boys"). Giulianelli et al. (2018) showed that the subject grammatical number is associated with various gates in an LSTM, and Lakretz et al. (2019) showed that ablation (disabling activation) of an LSTM model at certain locations can reduce its accuracy at scoring verbs of the correct grammatical number.

Influence offers an alternate means of exploring properties like number agreement. We say an input is *influential* on an outcome when changing just the input and nothing else induces a change on the outcome. In English grammar, the number of a subject is influential on the number of its verb, in that changing the number of that subject while keeping all other elements of a sentence fixed would necessitate a change in the number of the verb. Algorithmic transparency literature offers formal definitions for empirically quantifying notions of influence for systems in general (Datta et al., 2016) and for deep neural networks specifically (Leino et al., 2018; Sundararajan et al., 2017).

The mere fact that subject number is influential on verb number as output by an LSTM model is sufficient to conclude that it incorporates the agreement concept in some way but does not indicate whether it operates as a statistical heuristic or as a generalized rule. We address this question with *influence paths*, which decompose influence into a set of paths across the gates and neurons of an LSTM model. The approach has several elements:

1. Define an input parameter to vary the concept-specific quantity under study (e.g., the grammatical number of a particular noun, bottom-left node in Figure 1) and a concept-specific output feature to measure the parameter's effect on (e.g, number agreement with the parameterized noun, bottom-right node in Figure 1).

2. Apply a gradient-based influence method to quantify the influence of the concept parameter on the concept output feature; as per the chain rule, decompose the influence into model-path-specific quantities.

3. Inspect and characterize the distribution of influence across the model paths.

The paths demonstrate where relevant state information necessitated by the concept is kept, how it gets there, how it ends up being used to affect the model's output, and how and where related concepts interfere.

Our approach is state-agnostic in that it does not require *a priori* an assumption about how or if the concept will be implemented by the LSTM. This differs from works on diagnostic classifiers where a representation of the concept is assumed to exist in the network's latent space. The approach is also time-aware in that paths travel through cells/gates/neurons at different stages of an RNN evaluation. This differs from previous ablation-based techniques, which localize the number by clearing neurons at some position in an RNN for all time steps.

Our contributions are as follows:

- We introduce *influence paths*, a causal account of the use of concepts of interest as carried by paths across gates and neurons of an RNN.

- We demonstrate, using influence paths, that in a multi-layer LSTM language model, the concept of subject-verb number agreement is concentrated primarily on a single path (the red path in Figure 1), despite a variety of surrounding and intervening contexts.

- We show that attractors (intervening nouns of opposite number to the subject) do not diminish the contribution of the primary subject-verb path, but rather contribute their own influence of the opposite direction along the equivalent primary attractor-verb path (the blue path in the figure). This can lead to incorrect number prediction if an attractor's contribution overcomes the subject's.

- We corroborate and elaborate on existing results localizing subject number to the same two neurons which, in our results, lie on the primary path. We further extend and generalize prior compression/ablation results with a new path-focused compression test which verifies our localization conclusions.

Our results point to generalized knowledge as the answer to the central question. The number agreement concept is heavily centralized to the primary path despite the varieties of contexts. Further, the primary path's contribution is undiminished even amongst interfering contexts; number errors are not attributable to lack of the general number concept but rather to sufficiently influential contexts pushing the result in the opposite direction.

2 Background

LSTMs Long short-term memory networks (LSTMs) (Hochreiter and Schmidhuber, 1997) have proven to be effective for modeling sequences, such as language models, and empirically, this architecture has been found to be optimal compared to other second-order RNNs (Greff et al., 2017). LSTMs utilize several types of gates and internal states including *forget gates* (f), *input gates* (i), *output gates* (o), *cell states* (c), *candidate cell state* ($\tilde{c}$), and *hidden states* (h). Each gate is designed to carry out a certain function, or to fix a certain drawback of the vanilla RNN architecture. E.g., the forget gate is supposed to determine how much information from the previous cell state to retain or "forget", helping to fix the vanishing gradient problem (Hochreiter, 1998).

Number Agreement in Language Models The *number agreement* (NA) task, as described by Linzen et al. (2016), is an evaluation of a language model's ability to properly match the verb's grammatical number with its subject. This evaluation is performed on sentences specifically designed for the exercise, with zero or more words between the subject and the main verb, termed the *context*. The task for sentences with non-empty contexts will be referred to as *long-term* number agreement.

"Human-level" performance for this task can be achieved with a 2-layer LSTM language model (Gulordava et al.), indicating that the language model incorporates grammatical number despite being trained only for the more general word prediction task. Attempts to explain or localize the number concept within the model include (Lakretz et al., 2019), where ablation of neurons is applied to locate specific neurons where such information is stored; and (Giulianelli et al., 2018; Hupkes et al., 2018), where diagnostic classifiers are trained on gate activations to predict the number of the subject to see which gates or timesteps the number concept exhibits itself. These works also look at the special cases involving *attractors*—intervening nouns with grammatical number opposite to that of the subject (deemed instead *helpful nouns* if their number agrees with the subject)—such as the word "tree" in Figure 1. Both frameworks provide explanations as to why attractors lower the performance of NA tasks. However, they tend to focus on the activation patterns of gates or neurons without justifying their casual relationships with the concept of grammatical number, and do not explicitly identify the exact temporal trajectory of how the number of the subject influences the number of the verb.

Other relevant studies that look inside RNN models to locate specific linguistic concepts include visualization techniques such as (Karpathy et al., 2015), and explanations for supervised tasks involving LSTMs such as sentiment analysis (Murdoch et al., 2018).

Attribution Methods *Attribution methods* quantitatively measure the contribution of each of a function's individual inputs to its output. Gradient-based attribution methods compute the gradient of a model with respect to its inputs to describe how important each input is towards the output predictions. These methods have been applied to assist in explaining deep neural networks, predominantly in the image domain (Leino et al., 2018; Sundararajan et al., 2017; Bach et al., 2015; Simonyan et al., 2013). Some such methods are also axiomatically justified to provide a causal link between inputs (or intermediate neurons) and the output.

As a starting point in this work, we consider *Integrated Gradients* (IG) (Sundararajan et al., 2017). Given a *baseline*, x_0, the attribution for each input at point, x, is the path integral taken from the baseline to x of the gradients of the model's output with respect to its inputs. The baseline establishes a neutral point from which to make a counterfactual comparison; the attribution of a feature can be interpreted as the share of the model's output that is due to that feature deviating from its baseline value. By integrating the gradients along the linear interpolation from the baseline to x, IG ensures that the attribution given to each feature is *sensitive* to effects exhibited by the gradient at any point between the baseline and instance x.

Leino et al. (2018) generalize IG to better focus attribution on concepts other than just model outputs, by use of a *quantity of interest* (QoI) and a *distribution of interest* (DoI). Their measure, *Distributional Influence*, is given by Definition 1. The QoI is a function of the model's output expressing a particular output behavior of the model to calculate influence for; in IG, this is fixed as the model's output. The DoI specifies a distribution over which the influence should faithfully summarize the model's behavior; the influences are found by taking an expected value over DoI.

Definition 1 (Distributional Influence)**.** *With* quantity of interest*, q, and* distribution of interest*, D,*

the influence, χ, of the inputs on the quantity of interest is:

$$\chi(q, D) = \mathop{\mathbb{E}}_{\vec{x}\sim D}\left[\frac{\partial q}{\partial x}(\vec{x})\right]$$

The directed path integral used by IG can be implemented by setting the DoI to a uniform distribution over the line from the baseline to $\vec{x}$: $D = \text{Uniform}\left(\overline{\vec{x}_0\vec{x}}\right)$, for baseline, $\vec{x}_0$, and then multiplying χ by $\vec{x} - \vec{x}_0$. Conceptually, by multiplying by $\vec{x} - \vec{x_0}$, we are measuring the *attribution*, i.e., the contribution to the QoI, of $\vec{x} - \vec{x_0}$ by weighting its features by their *influence*. We use the framework of Leino et al. in this way to define our measure of attribution for NA tasks in Section 3.

Distributional Influence can be approximated by sampling according to the DoI. In particular, when using $D = \text{Uniform}\left(\overline{\vec{x}_0\vec{x}}\right)$ as noted above, Definition 1 can be computationally approximated with a sum of n intervals as in IG:

$$\chi \approx \sum_{i=1}^{n} \frac{\partial q}{\partial x}\left(\frac{i}{n}\vec{x} + \left(1 - \frac{i}{n}\right)\vec{x}_0\right)$$

Other related works include Fiacco et al. (2019), which employs the concept of neuron paths based on cofiring of neurons instead of influence, also on different NLP tasks from ours.

3 Methods

Our method for computing influence paths begins with modeling a relevant concept, such as grammatical number, in the influence framework of Leino et al. (Definition 1) by defining a quantity of interest that corresponds to the grammatical number of the verb, and defining a component of the input embedding that isolates the subject's grammatical number (Section 3.1). We then decompose the influence measure along the relevant structures of LSTM (gates or neurons) as per standard calculus identities to obtain a definition for *influence paths* (Section 3.2).

3.1 Measuring Number Agreement

For the NA task, we view the initial fragment containing the subject as the input, and the word distribution at the position of its corresponding verb as the output.

Formally, each instance in this task is a sequence of d-dimensional word embedding vectors, $\mathbf{w} \stackrel{\text{def}}{=} \langle \vec{w}_i \rangle_i$, containing the subject and the corresponding verb, potentially with intervening words in between. We assume the subject is at position t and the verb at position $t + n$. The output score of a word, w, at position i will be written $s_i(w)$. If w has a grammatical number, we write w^+ and w^- to designate w with its original number and the equivalent word with the opposite number, respectively.

Quantity of Interest We instrument the output score with a QoI measuring the agreement of the output's grammatical number to that of the subject:

Definition 2 (Number Agreement Measure)**.** *Given a sentence,* $\mathbf{w}$*, with verb,* w*, whose correct form (w.r.t. grammatical number) is* w^+*, the* quantity of interest*,* q*, measures the correctness of the grammatical number of the verb:*

$$q\left(\mathbf{w}\right) \stackrel{\text{def}}{=} s_{t+n}\left(w^+\right) - s_{t+n}\left(w^-\right)$$

In plain English, q captures the weight that the model assigns to the correct form of w as opposed to the weight it places on the incorrect form. Note that the number agreement concept could have reasonably been measured using a different quantity of interest. E.g., considering the scores of *all* vocabulary words of the correct number and incorrect number in the positive and negative terms, respectively, is an another alternative. However, based on our preliminary experiments, we found this alternative does not result in meaningful changes to the reported results in the further sections.

Distribution of Interest We also define a component of the embedding of the subject that captures its grammatical number, and a distribution over the inputs that allows us to sensitively measure the influence of this concept on our chosen quantity of interest. Let $\vec{w}^0$ be the word embedding midway between its numbered variants, i.e., $\frac{\vec{w}^+ + \vec{w}^-}{2}$. Though this vector will typically not correspond to any English word, we interpret it as a number-neutral version of $\vec{w}$. Various works show that linear arithmetic on word embeddings of this sort preserves meaningful word semantics as demonstrated in analogy parallelograms (Mikolov et al., 2013). Finally, given a sentence, $\mathbf{w}$, let $\mathbf{w}_t^0$ be the sentence $\mathbf{w}$, except with the word embedding $\vec{w}_t$ replaced with its neutral form $\vec{w}_t^0$. We see that $\mathbf{w} - \mathbf{w}_t^0$ captures the part of the input corresponding to the grammatical number of the subject, $\vec{w}_t$.

Definition 3 (Grammatical Number Distribution)**.** *Given a singular (or plural) noun,* w_t*, in a sentence,* $\mathbf{w}$*, the distribution density of sentences,* $D_{\mathbf{w}}$*, exercising the noun's* singularity *(or* plurality*) linearly*

interpolates between the neutral sentence, $\mathbf{w}_t^0$*, and the given sentence,* $\mathbf{w}$*:*

$$D_{\mathbf{w}} \stackrel{\text{def}}{=} \textit{Uniform}\left(\overline{\mathbf{w}_t^0 \mathbf{w}}\right)$$

If $\vec{w}_t$ is singular, our counterfactual sentences span $\mathbf{w}$ with number-neutral $\vec{w}_t^0$ all the way to its singular form $\vec{w}_t = \vec{w}_t^+$. We thus call this distribution a *singularity* distribution. Were w_t plural instead, we would refer to the distribution as a *plurality* distribution. Using this distribution of sentences as our DoI thus allows us to measure the influence of $\mathbf{w} - \mathbf{w}_t^0$ (the grammatical number of a noun at position t) on our quantity of interest *sensitively* (in the sense that Sundararajan et al. define their axiom of sensitivity for IG (Sundararajan et al., 2017)).

Subject-Verb Number Agreement Putting things together, we define our attribution measure.

Definition 4 (Subject-Verb Number Agreement Attribution)**.** *The measure of attribution,* α*, of a noun's grammatical number on the subject-verb number agreement is defined in terms of the DoI,* $D_{\mathbf{w}}$*, and QoI,* q*, as in Definitions 3 and 2, respectively.*

$$\alpha(\mathbf{w}) = (\mathbf{w} - \mathbf{w}_t^0)\, \chi(q, D_{\mathbf{w}})$$

Essentially, the attribution measure weights the features of the subject's grammatical number by their Distributional Influence, χ. Because $D_{\mathbf{w}}$ is a uniform distribution over the line segment between $\mathbf{w}$ and $\mathbf{w}_t^0$, as with IG, the attribution can be interpreted as each feature's net contribution to the change in the QoI, $q(\mathbf{w}) - q(\mathbf{w}_t^0)$, as $\sum_i \chi(\mathbf{w})_i = q(\mathbf{w}) - q(\mathbf{w}_t^0)$ (i.e., Definition 4 satisfies the axiom Sundararajan et al. term *completeness* (Sundararajan et al., 2017)).

In Figure 1, for instance, this definition measures the attribution from the plurality of the subject ("boys"), towards the model's prediction of the correctly numbered verb ("run") versus the incorrectly numbered verb ("runs"). Later in this paper we will also investigate the attribution of intervening nouns on this same quantity. We expect the input attribution to be positive for all subjects and helpful nouns, and negative for attractors, which can be verified by the P^+columns of Table 1 (the details of this experiment are introduced in Section 4).

3.2 Influence Paths

Input attribution as defined by IG (Sundararajan et al., 2017) provides a way of explaining a model by highlighting the input dimensions with large attribution towards the output. Distributional Influence (Leino et al., 2018) with a carefully chosen QoI and DoI (Definition 4) further focuses the influence on a concept at hand, grammatical number agreement. Neither, however, demonstrate how these measures are conveyed by the inner workings of a model. In this section we define a decomposition of the influence into paths of a model, thereby assigning attribution not just to inputs, but also to the internal structures of a given model.

We first define arbitrary deep learning models as computational graphs, as in Definition 5. We then use this graph abstraction to define a notion of influence for a path through the graph. We posit that any natural path decomposition should satisfy the following conservation property: *the sum of the influence of each path from the input to the output should equal the influence of the input on the QoI.* We then observe that the chain rule from calculus offers one such natural decomposition, yielding Definition 6.

Definition 5 (Model)**.** *A model is an acyclic graph with a set of nodes, edges, and activation functions associated with each node. The output of a node,* n*, on input* x *is* $n(x) \stackrel{\text{def}}{=} f_n(n_1(x), \cdots, n_m(x))$ *where* $n_1, \cdots, n_m$ *are* n*'s predecessors and* f_n *is its activation function. If* n *does not have predecessors (it is an input), its activation is* $f_n(x)$*. We assume that the domains and ranges of all activation functions are real vectors of arbitrary dimension.*

We will write $n_1 \to n_2$ *to denote an edge (i.e.,* n_1 *is a direct predecessor of* n_2*), and* $n_1 \to^* n_2$ *to denote the set of all paths from* n_1 *to* n_2*. The partial derivative of the activation of* n_2 *with respect to the activation of* n_1 *will be written* $\frac{\partial n_2}{\partial n_1}$*.*

This view of a computation model is an extension of network decompositions from attribution methods using the natural concept of "layers" or "slices" (Dhamdhere et al., 2018; Leino et al., 2018; Bach et al., 2015). This decomposition can be tailored to the level of granularity we wish to expose. Moreover, in RNN models where no single and consistent "natural layer" can be found due to the variable-length inputs, a more general graph view provides the necessary versatility.

Definition 6 (Path Influence)**.** *Expanding Definition 4 using the chain rule, the influence of input*

node, s, on target node, t, in a model, G, is:

$$\chi_s = \mathop{\mathbb{E}}_{x\sim D(x)}\left[\frac{\partial t}{\partial s}(x)\right]$$
$$= \mathop{\mathbb{E}}_{x\sim D(x)}\left[\sum_{p\in(s\rightarrow^* t)}\prod_{(n_1\rightarrow n_2)\in p}\frac{\partial n_2}{\partial n_1}(x)\right]$$
$$= \sum_{p\in(s\rightarrow^* t)}\underbrace{\mathop{\mathbb{E}}_{x\sim D(x)}\left[\prod_{(n_1\rightarrow n_2)\in p}\frac{\partial n_2}{\partial n_1}(x)\right]}_{\chi_s^p}$$

Note that the same LSTM can be modeled with different graphs to achieve a desired level of abstraction. We will use two particular levels of granularity: a coarse *gate-level* abstraction where nodes are LSTM gates, and a fine *neuron-level* abstraction where nodes are the vector elements of those gates. Though the choice of abstraction granularity has no effect on the represented model semantics, it has implications on graph paths and the scale of their individual contributions in a model.

Gate-level and Neuron-level Paths We define the set of gate-level nodes to include: $\{f_t^l,\ i_t^l,\ o_t^l,\ c_t^l,\ \tilde{c}_t^l,\ h_t^l\ :\ t<T,\ l<L\}$, where T is the number of time steps (words) and L is number of LSTM layers. The node set also includes an attribution-specific input node ($\mathbf{w}-\mathbf{w}_t^0$) and an output node (the QoI). An example of this is illustrated in Figure 2. We exclude intermediate calculations (the solid nodes of Figure 2, such as $f_t\odot c_{t-1}$) as their inclusion does not change the set of paths in a graph. We can also break down each vector node into scalar components and further decompose the gate-level model into a neuron-level one: $\{f_{ti}^l,\ i_{ti}^l,\ o_{ti}^l,\ c_{ti}^l,\ \tilde{c}_{ti}^l,\ h_{ti}^l\ :\ t<T,\ i<H,\ l<L\}$, where H is the size of each gate vector. This decomposition results in an exponentially large number of paths. However, since many functions between gates in an LSTM are element-wise operations, neuron-level connections between many neighboring gates are sparse.

Path Refinement While the neuron-level path decomposition can theoretically be performed on the whole network, in practice we choose to specify a gate-level path first, then further decompose that path into neuron-level paths. We also collapse selected vector nodes, allowing us to further localize a concept on a neuron level while avoiding an explosion in the number of paths. The effect of this pipeline will be empirically justified in Section 4.

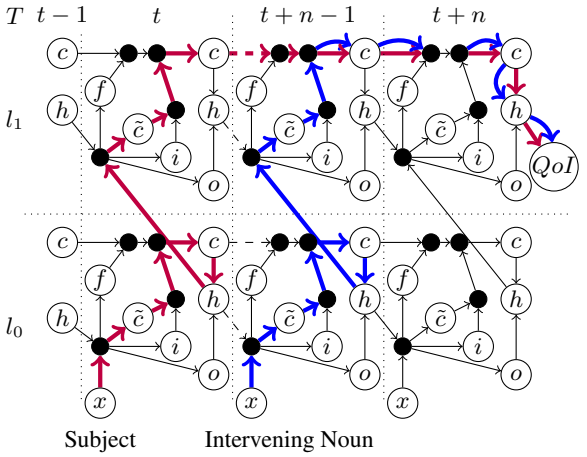

Figure 2: Influence path diagram in a NA task for the 2-layer LSTM model. The red path shows the path with the greatest attribution (the primary path) from the subject; The blue path shows the primary path from the intervening noun.

4 Evaluation

In this section we apply influence path decomposition to the NA task. We investigate major gate-level paths and their influence concentrations in Section 4.2. We further show the relations between these paths and the paths carrying grammatical number from intervening nouns (i.e. attractors & helpful nouns) in Section 4.3. In both we also investigate high-attribution neurons along primary paths allowing us to compare our results to prior work.

4.1 Dataset and Model

We study the exact combination of language model and NA datasets used in the closely related prior work of Lakretz et al. (2019). The pre-trained language model of Gulordava et al. and Lakretz et al. is a 2-layer LSTM trained from Wikipedia articles. The number agreement datasets of Lakretz et al. are several synthetically generated datasets varying in syntactic structures and in the number of nouns between the subject and verb.

For example, *nounPP* refers to sentences containing a noun subject followed by a prepositional phrase such as in Figure 1. Each NA task has subject number (and intervening noun number if present) realizations along singular (S) and plural (P) forms. In listings we denote subject number (S or P) first and additional noun (if any) number second. Details including the accuracy of the model on the NA tasks are summarized by Lakretz et al. (2019). Our evaluation replicates part of Table 2 in

said work.

4.2 Decomposing Number Agreement

We begin with the attribution of subject number on its corresponding verb, as decomposed per Definition 6. Among all NA tasks, the gate-level path carrying the most attribution is one following the same pattern with differences only in the size of contexts. With indices t and $t+n$ referring to the subject and verb respectively, this path, which we term the *primary path of subject-verb number agreement*, is as follows:

$$x_t(DoI) \cdot \tilde{c}^0 \cdot c^0 \cdot h^0 \cdot \tilde{c}^1 \cdot \left(c^1\right)^* \cdot h^1 \cdot QoI$$

The primary path is represented by the red path in Figure 2. The influence first passes through the temporary cell state $\tilde{c}^0$, the only non-sigmoid cell states capable of storing more information than sigmoid gates, since $i, f, o \in (0, 1)$ while the $tanh$ gate $\tilde{c} \in (-1, 1)$. Then the path passes through c^0, h^0, and similarly to c^1 through $\tilde{c}^1$, jumping from the first to the second layer. The path then stays at c^1, through the direct connections between cell states of neighbouring time steps, as though it is "stored" there without any interference from subsequent words. As a result, this path is intuitively the most efficient and simplistic way for the model to encode and store a "number bit."

The extent to which this path can be viewed as *primary* is measured by two metrics. The results across a subset of syntactic structures and number conditions mirroring those in Lakretz et al. (2019) are shown in Table 1. We include 3 representative variations of the task. The metrics are:

1. t-value: probability that a given path has greater attribution than a uniformly sampled path on a uniformly sampled sentence.

2. Positive/Negative Share ($\pm$Share): expected (over sentences) fraction of total positive (or negative) attribution assigned to the given positive (or negative) path.

Per Table 1 (From Subject, Primary Path), we make our first main observation:

Observation 1. *The same one primary path consistently carries the largest amount positive attribution across all contexts as compared to all other paths.*

Even in the case of its smallest share (nounPPAdv), the 3% share is large when taking into account more than 40,000 paths in total. Sentences with singular subjects (top part of Table 1) have a slightly stronger concentration of attribution in the primary path than plural subjects (bottom part of Table 1), possibly due to English plural (infinitive) verb forms occurring more frequently than singular forms, thus less concentration of attribution is needed due to the "default signal" in place.

Primary Neurons We further decompose the primary path into influence passing through each neuron. Since only connections between second layer cell states are sparse, we only decompose the segment of the primary path from c_t^1 to c_{t+n}^1, resulting in a total of 650 (the number of hidden units) neuron-level paths. (We leave the non-sparse decompositions for future work). The path for neuron i, for example, is represented as:

$$x_t(DoI) \cdot \tilde{c}^0 \cdot c^0 \cdot h^0 \cdot \tilde{c}^1 \cdot \left(c_i^1\right)^* \cdot h^1 \cdot QoI$$

To compare the attribution of an individual neuron with all other neurons, we employ a similar aforementioned t-value, where each neuron-level path is compared against other neuron-level paths.

The results of the neuron-level analysis are shown in Table 1 (From Subject, Primary Neuron). Out of the 650 neuron-level paths in the gate-level primary path, we discover two neurons with consistently the most attribution (neurons 125 and 337 of the second layer). This indicates the number concept is concentrated in only two neurons.

Comparison with Lakretz et al. (2019) Unco-incidentally, both neurons match the units found through ablation by Lakretz et al., who use the same model and dataset (neurons 988 and 776 are neurons 125 and 337 of the second layer). This accordance to some extent verifies that the neurons found through influence paths are functionally important. However, the t-values shown in Table 1 show that both neuron 125 and 337 are influential regardless of the subject number, whereas Lakretz et al. assign a subject number for each of these two neurons due to their disparate effect in lowering accuracy in ablation experiments. One possible reason is that the ablation mechanism used in (Lakretz et al., 2019) assumes that a "neutral number state" can be represented by zero-activations for all gates, while in reality the network may encode the neutral state differently for different gates.

Another major distinction of our analysis from Lakretz et al. (2019) regards *simple* cases with no

Task	C	From Subject						From Intervening Noun					
		P_+	$\|P\|$	Primary Path		Primary Neuron		P_+	$\|P\|$	Primary Path		Primary Neuron	
				+Share	t	t_{125}	t_{337}			$\pm$ Share	t	t_{125}	t_{337}
Simple	S	1.0	16	0.47	1.0	0.99	1.0	-	-	-	-	-	-
nounPP	SS	1.0	6946	0.1	1.0	1.0	1.0	0.82	16	0.31(+)	0.9	0.78	0.98
nounPP	SP	1.0	6946	0.1	1.0	1.0	1.0	0.23	16	0.24(-)	0.23	0.06	0.15
nounPPAdv	SS	1.0	41561	0.07	1.0	1.0	1.0	0.92	152	0.09(+)	0.96	0.85	1.0
nounPPAdv	SP	1.0	41561	0.07	1.0	1.0	1.0	0.32	152	0.09(-)	0.14	0.13	0.01
Simple	P	1.0	16	0.33	0.93	0.97	0.99	-	-	-	-	-	-
nounPP	PS	1.0	6946	0.05	0.91	0.99	1.0	0.06	16	0.28(-)	0.21	0.22	0.12
nounPP	PP	1.0	6946	0.05	0.92	0.99	1.0	0.95	16	0.31(+)	0.9	0.97	0.79
nounPPAdv	PS	1.0	41561	0.03	0.93	0.99	1.0	0.32	152	0.04(-)	0.28	0.41	0.16
nounPPAdv	PP	1.0	41561	0.03	0.92	0.99	1.0	0.83	152	0.07(+)	0.92	0.99	0.84

Table 1: Statistics for attribution of primary paths and neurons from the subject/intervening noun: P_+ is the percentage of sentences with positive input attribution. Task and C columns refer to sentence structures in Lakretz et al. (2019). $|P|$ is the total number of paths; t and $\pm$Share are t-values and positive/negative share, respectively. For calculating t_{125} and t_{337} of primary neurons (125 and 337), we exclude these two neurons to avoid comparing them with each other.

word between subjects and verbs. Unlike Lakretz et al., who claim that the two identified neurons are "long-term neurons", we discover that these two neurons are also the only neurons important for short-term number agreement. This localization cannot be achieved by diagnostic classifiers used by Lakretz et al., indicating that the signal can be better uncovered using influence-based paths rather than association-based methods such as ablation.

4.3 Decomposing from Intervening Nouns

Next we focus on NA tasks with intervening nouns and make the following observation:

Observation 2. *The primary subject-verb path still accounts for the largest positive attribution in contexts with either attractors or helpful nouns.*

A slightly worse NA task performance (Lakretz et al., 2019) in cases of attractors (SP, PS) indicates that they interfere with prediction of the correct verb. In contrast, we also observe that helpful nouns (SS, PP) contribute positively to the correct verb number (although they should not from a grammar perspective).

Primary Path from the Intervening Noun We adapt our number agreement concept (Definition 2) by focusing the DoI on the intervening noun, thereby allowing us to decompose its influence on the verb number not grammatically associated with it. In Table 1 (From Intervening Noun) we discover a similar primary path from the intervening noun:

Observation 3. *Attribution towards verb number from intervening nouns follows the same primary path as the subject but is of lower magnitude and reflects either positive or negative attribution in cases of helpful nouns or attractors, respectively.*

Task	C	Compression Scheme						
		$\overline{C_{si}}$	$\overline{C_s}$	$\overline{C_i}$	C_{si}	C_s	C_i	C
nounPP	SS	.66	.77	.95	.93	.71	.77	.95
nounPP	SP	.64	.36	.94	.64	.75	.40	.74
nounPP	PS	.34	.24	.92	.40	.69	.18	.80
nounPP	PP	.39	.66	.91	.76	.68	.58	.97
nounPP	mean	.51	.51	.93	.68	.70	.48	.87
nounPPAdv	SS	.70	.86	.98	.73	.56	.43	1.0
nounPPAdv	SP	.70	.43	.99	.50	.60	.27	.88
nounPPAdv	PS	.38	.22	.98	.76	.79	.56	.96
nounPPAdv	PP	.39	.67	.98	.84	.83	.76	1.0
nounPPAdv	mean	.54	.55	.99	.71	.69	.50	.96

Table 2: Model compression accuracy under various compression schemes. C is the uncompressed model.

This disparity in magnitude is expected since the language model possibly identifies the subject as the head noun through the prepositions such as "behind" in Figure 1, while still needing to track the number of the intervening noun in possible clausal structures. Such need is comparably weaker compared to tracking numbers of subjects, possibly because in English, intervening clauses are rarer than intervening non-clauses. Similar arguments can be made for neuron-level paths.

4.4 Model Compression

Though the primary paths are the highest contributors to NA tasks, it is possible that collections of associated non-primary paths account for more of the verb number concept. We gauge the extent to which the primary paths alone are responsible for the concept with compression/ablation exper-

iments. We show that the computations relevant to a specific path alone are sufficient in maintaining performance for the NA task. We *compress* the model by specifying node sets to preserve, and intervene on the activations of all other nodes by setting their activations to constant expected values (average over all samples). We choose the expected values instead of full ablation (setting them to zero), as ablation would nullify the function of Sigmoid gates. For example, to compress the model down to the red path in Figure 2, we only calculate the activation for gates $\tilde{c}_t^0$ and $\tilde{c}_t^1$ for each sample, while setting the activation of all other $\tilde{c}, f, o, i$ to their average values over all samples. In Table 2, we list variations of the compression schemes based on the following preserved node sets:

$$C \stackrel{\text{def}}{=} \left\{ f_t^l, i_t^l, o_t^l, \tilde{c}_t^l : t_{\text{sub}} < t < t_{\text{verb}}, l \in \{0, 1\} \right\}$$
$$C_s \stackrel{\text{def}}{=} \left\{ \tilde{c}_{t_{\text{sub}}}^0, \tilde{c}_{t_{\text{sub}}}^1 \right\} \quad C_i \stackrel{\text{def}}{=} \left\{ \tilde{c}_{t_{\text{int}}}^0, \tilde{c}_{t_{\text{int}}}^1 \right\}$$
$$C_{si} \stackrel{\text{def}}{=} C_s \cup C_i$$

For example, column C_{si} in Table 2 shows the accuracy when the compressed model only retains the primary path from both the subject and the intervening noun while the computations of all other paths are set to their expected values; while in $\overline{C_{si}}$, all paths but the paths in C_{si} are kept.

We observe that the best compressed model is $\overline{C_i}$, where the primary path from the intervening noun is left out; it performs even better than the original model; the increase comes from the cases with attractors (PS, SP). This indicates that eliminating the primary path from the attractor improves the model. The next best models apart from C are C_s and C_{si}, where primary paths are kept. Compressed models without the primary subject-verb path ($\overline{C_{si}}$, $\overline{C_s}$, C_i) have performances close to random guessing.

Observation 4. *Accuracy under path-based model compression tests corroborate that primary paths account for most of the subject number agreement concept of the LSTM.*

By comparing the SP and PS rows of $\overline{C_{si}}$, $\overline{C_s}$, C_s, and C_i, we observe the effect of attractors in misguiding the model into giving wrong predictions. Similarly, we see that helpful nouns (SS, PP) help guide the models to make more accurate predictions, though this is not grammatically justified.

5 Conclusions

The combination of finely-tuned attribution and gradient decomposition lets us investigate the handling of the grammatical number agreement concept attributed to paths across LSTM components. The concentration of attribution to a primary path and two primary cell state neurons and its persistence in a variety of short-term and long-term contexts, even with confounding attractors, demonstrates that the concept's handling is, to a large degree, general and localized. Though the heuristic decisioning aspect of an LSTM is present in the large quantities of paths with non-zero influence, their overall contribution to the concept is insignificant as compared to the primary path. Node-based compression results further corroborate these conclusions.

We note, however, that our results are based on datasets exercising the agreement concept in contexts of a limited size. We speculate that the primary path's attribution diminishes with the length of the context, which would suggest that at some context size, the handling of number will devolve to be mostly heuristic-like with no significant primary paths. Though our present datasets do not pose computational problems, the number of paths, at both the neuron and the gate level, is exponential with respect to context size. Investigating longer contexts, the diminishing dominance of the primary path, and the requisite algorithmic scalability requirements are elements of our ongoing work.

We also note that our method can be expanded to explore number agreement in more complicated sentences with clausal structures, or other syntactic/semantic signals such as coreference or gender agreement.

Acknowledgement This work was developed with the support of NSF grant CNS-1704845 as well as by DARPA and the Air Force Research Laboratory under agreement number FA8750-15-2-0277. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes not withstanding any copyright notation thereon. The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of DARPA, the Air Force Research Laboratory, the National Science Foundation, or the U.S. Government. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for this work.

References

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

Anupam Datta, Shayak Sen, and Yair Zick. 2016. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE symposium on security and privacy (SP)*, pages 598–617. IEEE.

Kedar Dhamdhere, Mukund Sundararajan, and Qiqi Yan. 2018. How important is a neuron? *arXiv preprint arXiv:1805.12233*.

James Fiacco, Samridhi Choudhary, and Carolyn Rose. 2019. Deep neural model inspection and comparison via functional neuron pathways. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5754–5764.

F. A. Gers and E. Schmidhuber. 2001. Lstm recurrent networks learn simple context-free and context-sensitive languages. *Trans. Neur. Netw.*, 12(6):1333–1340.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. Under the Hood: Using Diagnostic Classifiers to Investigate and Improve how Language Models Track Agreement Information.

Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. 2017. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. Colorless green recurrent networks dream hierarchically.

Sepp Hochreiter. 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926.

Andrej Karpathy, Justin Johnson, and Li Fei-Fei. 2015. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*.

Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. 2019. The emergence of number and syntax units in lstm language models. *arXiv preprint arXiv:1903.07435*.

Klas Leino, Shayak Sen, Anupam Datta, Matt Fredrikson, and Linyi Li. 2018. Influence-directed explanations for deep convolutional networks. In *2018 IEEE International Test Conference (ITC)*, pages 1–8. IEEE.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

W James Murdoch, Peter J Liu, and Bin Yu. 2018. Beyond word importance: Contextual decomposition to extract interactions from lstms. *arXiv preprint arXiv:1801.05453*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org.