MAPPING THE TIMESCALE ORGANIZATION OF NEURAL LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

In the human brain, sequences of language input are processed within a distributed and hierarchical architecture, in which higher stages of processing encode contextual information over longer timescales. In contrast, in recurrent neural networks which perform natural language processing, we know little about how the multiple timescales of contextual information are functionally organized. Therefore, we applied tools developed in neuroscience to map the "processing timescales" of individual units within a word-level LSTM language model. This timescale-mapping method assigned long timescales to units previously found to track long-range syntactic dependencies, and revealed a new cluster of previously unreported long-timescale units. Next, we explored the functional role of units by examining the relationship between their processing timescales and network connectivity. We identified two classes of long-timescale units: "Controller" units composed a densely interconnected subnetwork and strongly projected to the forget and input gates of the rest of the network, while "Integrator" units showed the longest timescales in the network, and expressed projection profiles closer to the mean projection profile. Ablating integrator and controller units affected model performance at different position of a sentence, suggesting distinctive functions of these two sets of units. Finally, we tested the generalization of these results to a character-level LSTM model. In summary, we demonstrated a model-free technique for mapping the timescale organization in neural network models, and we applied this method to reveal the timescale and functional organization of LSTM language models.[1]

1 INTRODUCTION

Language processing requires tracking information over multiple timescales. To be able to predict the final word "timescales" in the previous sentence, one must consider both the short-range context (e.g. the adjective "multiple") and the long-range context (e.g. the subject "language processing"). How do humans and neural language models encode such multi-scale context information? Neuroscientists have developed methods to study how the human brain encodes information over multiple timescales during sequence processing. By parametrically varying the timescale of intact context, and measuring changes in the neural response, a series of studies (Lerner et al., 2011; Xu et al., 2005; Honey et al., 2012) showed that higher-order regions are more sensitive to long-range context change than lower-order sensory regions. These studies indicate the existence of a "hierarchy of processing timescales" in the human brain. More recently, Chien & Honey (2020) used a time-resolved method to investigate how the brain builds shared context, when two groups of people processed the same sentence preceded by different contexts. By directly mapping the time required for individual brain regions to converge on a shared representation in response to shared input, they confirmed that higher-order regions take longer to converge to a shared context representation. Altogether, these and other lines of investigation suggest that sequence processing in the brain is supported by a distributed and hierarchical structure: sensory regions have short processing timescales and are primarily influenced by the current input and its short-range context, while higher-order cortical regions have longer timescales and track longer-range dependencies (Hasson et al., 2015; Honey et al., 2012; Chien & Honey, 2020; Lerner et al., 2011; Baldassano et al., 2017; Runyan et al., 2017; Fuster, 1997).

[1]The code was shared via [this link]

How are processing timescales organized within recurrent neural networks (RNNs) trained to perform natural language processing? Long short-term memory networks (LSTMs) (Hochreiter & Schmidhuber, 1997) have been widely investigated in terms of their ability to successfully solve sequential prediction tasks. However, long-range dependencies have usually been studied with respect to a particular linguistic function (e.g. subject-verb number agreement (Linzen et al., 2016; Gulordava et al., 2018; Lakretz et al., 2019), and there has been less attention on the broader question of how sensitivity to prior context – broadly construed – is functionally organized within these recurrent networks. Therefore, drawing on prior work in the neuroscience literature, here we demonstrate a model-free approach to mapping processing timescale in recurrent neural networks. We focused on existing language models that were trained to predict upcoming tokens at the word level (Gulordava et al., 2018) and at the character level (Hahn & Baroni, 2019). The timescale organization of these two models both revealed that the higher layers of LSTM language models contained a small subset of units which exhibit long-range sequence dependencies; this subset includes previously reported units (e.g. a "syntax" unit, Lakretz et al., 2019) as well as previously unreported units.

After mapping the timescales of individual units, we asked: do the processing timescales of each unit in the network relate to its functional role, as measured by its connectivity? The question is motivated by neuroscience studies which have shown that in the human brain, higher-degree nodes tend to exhibit slower dynamics and longer context dependence than lower-degree nodes (Baria et al., 2013). More generally, the primate brain exhibits a core periphery structure in which a relatively small number of "higher order" and high-degree regions (in the prefrontal cortex, in default-mode regions and in so-called "limbic" zones) maintain a large number of connections with one another, and exert a powerful influence over large-scale cortical dynamics (Hagmann et al., 2008; Mesulam, 1998; Gu et al., 2015). Inspired by the relationships between timescales and network structure in neuroscience, we set out to test corresponding hypotheses in neural nets: (1) Do units with longer-timescales tend to have higher degree in neural language models? and (2) Do neural language models also exhibit a "core network" composed of functionally influential high-degree units? Using an exploratory network-theoretic approach, we found that units with longer timescales tend to have more projections to other units. Furthermore, we identified a set of medium-to-long timescale "controller" units which exhibit distinct and strong projections to the gating nodes that control the state of other units, and a set of long-timescale "integrator units" which showed influence on predicting words where the long context is relevant. In summary, these findings advance our understanding of the timescale distribution and functional organization of LSTM language models, and provide a method for identifying the subset of important nodes for representing long-range contextual information.

2 Related Work

Linguistic Context in LSTMs How do LSTMs encode linguistic context at multiple timescales? Prior work suggested that the units sensitive to information that requires long-range dependencies (such as number agreement) are sparse. By ablating one unit at a time, Lakretz et al. (2019) found two units that encode information required for processing long-range subject-verb number agreement (one for singular and one for plural information encoding). They further identified several long-range "syntax units" whose activation was associated with syntactic tree-depth. Overall, Lakretz et al. (2019) suggests that a sparse subset of units tracks long-range dependencies related to subject-verb agreement and syntax. If this pattern is general – i.e. if there are very few nodes tracking long-range dependencies in general – this may limit the capacity of the models to process long sentences with high complexity, for reasons similar to those that may limit human sentence processing (Lakretz et al., 2020). To test whether long-range nodes are sparse in general, we require a model-free approach for mapping the context dependencies of every unit in the language network.

Whole-network context dependence. Previous work by Khandelwal et al. (2018) investigated the duration of prior context that LSTM language models use to support word prediction. Context-dependence was measured by permuting the order of words preceding the preserved context, and observing the increase in model perplexity when the preserved context gets shorter. Khandelwal et al. (2018) found that up to 200 word-tokens of prior context were relevant to the model perplexity, but that the precise ordering of words only mattered within the most recent 50 tokens. The context-permutation method employed in this study was analogous to the approach used to measure context-dependence in human brain responses to visual movies (Hasson et al., 2008) and to auditory language Lerner et al. (2011).

Inspired by the findings of (Khandelwal et al., 2018) and (Lakretz et al., 2019), in the present study we set out to map the context-dependence across all of the individual units in the LSTM model. This enabled us to relate the timescales to the effects of node-specific ablation and the network architecture itself. In addition, our context manipulations included both context-swapping (substituting alternative meaningful contexts) and context-shuffling (permuting the words in the prior context to disrupt inter-word structure), which allowed us to better understand how individual words and syntactically structured word-sequences contribute to the the context representation of individual hidden units.

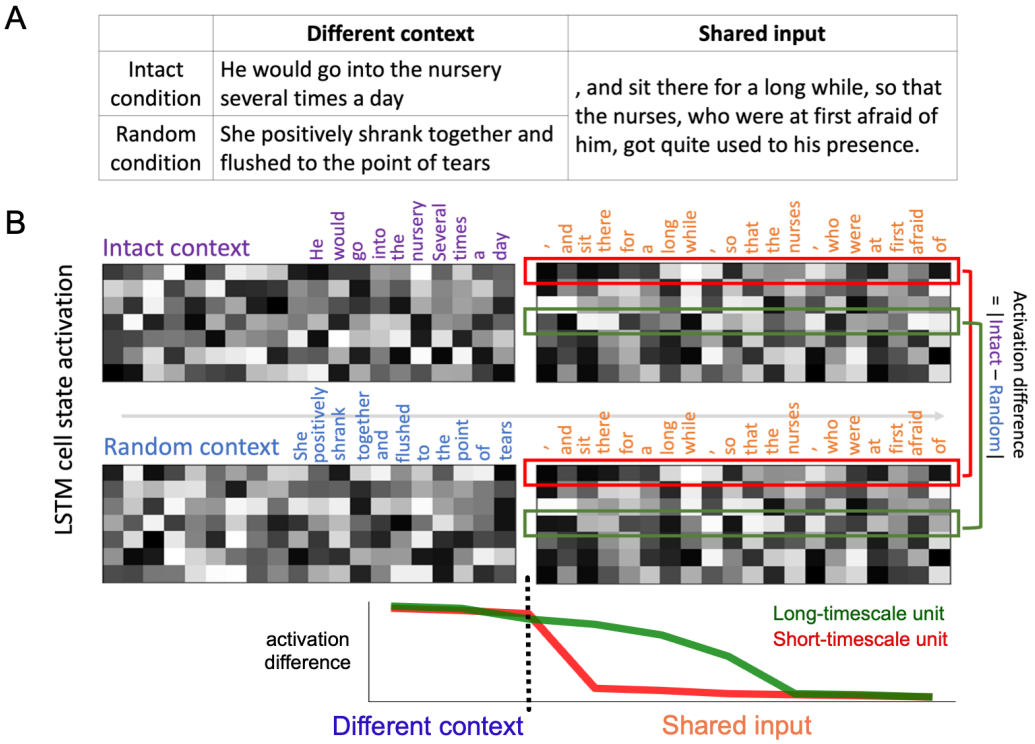

Figure 1: Context dependence paradigm for mapping processing timescales of individual units. **A.** Example sentences for the model to process in the Intact Context and Random Context condition. In the Intact Context condition, the context segment was followed by the shared segment from the novel, while in the Random Context condition, the context segment is replaced by randomly sampled segments. **B.** Schematic hidden state representation of the neural network. When the model is processing different context segments, the hidden unit activation is different; when the model starts to process the shared segment, the hidden unit activation difference decreases over time with different rates. The activation difference is the absolute difference of unit activation between the two conditions. The expected decreasing pattern of activation difference of a long-timescale unit and a short-timescale unit are shown in green and red, respectively.

3 Methods

3.1 Language Models and Corpus

We evaluated the internal representations generated by a pre-trained word-level LSTM language model (WLSTM Gulordava et al., 2018) as well as a pre-trained character-level LSTM model (CLSTM, Hahn & Baroni, 2019) as they processed sentences sampled from the 427804-word/1965719-character novel corpus: *Anna Karenina* by Leo Tolstoy [2] [3].

[2](Tolstoy, 2016) translated from Russian to English by Constance Garnett.

[3]retrieved from http://www.gutenberg.org/ebooks/1399

For WLSTM, we used the model made available by Gulordava et al. (2018). The model has a 650-dimensional embedding layer, two 650-dimensional hidden layers and an output layer with vocabulary size 50,000. The model was trained and tested on Wikipedia sentences and was not fine-tuned to the novel corpus. Therefore, we only used sentences with low perplexity from the novel in our main timescale analysis. We did the same analysis using the Wikipedia test set from Gulordava et al. (2018) and yielded similar results (See Section 5.3, Figure A.4A, Appendix A.2.1).

For CLSTM, we used the model made available by Hahn & Baroni (2019). The model has a 200-dimensional embedding layer, three 1024-dimensional hidden layers and an output layer with vocabulary size 63. The model was trained on Wikipedia data with all characters lower-cased and whitespace removed. We tested the model with sentences sampled from *Anna Karenina* as the WLSTM model, and we got bits-per-character (BPC) close to what Hahn & Baroni (2019) reported in the original paper.

3.2 Temporal Context Construction Paradigm

In order to determine the processing timescales of cell state vectors and individual units, we modified the "temporal context construction" method developed by Chien & Honey (2020). Thus, the internal representations of the model were compared across two conditions: (1) the Intact Context condition and (2) the Random Context condition. In both conditions, the model was processing the same shared sequence of words (for example, segment B), but the preceding sentence differed across the two conditions. In the Intact Context condition, the model processed segment B (the shared segment) preceded by segment A, which was the actual preceding context from the original text. In the current study, for example, segment A and B are connected by ", and" within long sentences from the novel corpus (Figure 1A), to ensure the temporal dependencies between A and B. In the Random Context condition, however, the model processed the same shared input (segment B), but the context was replaced by segment X, which was a randomly sampled segment from the rest of the corpus. Segment X was therefore not usually coherently related to segment B. For the WLSTM timescale analysis, we chose long sentences in the Intact Context condition that satisfied the following constraints: (1) mean perplexity across all words in the sentence < 200, (2) the shared segment was longer than 25 words, and (3) the context segment was longer than 10 words. 77 sentences are included as trials in our analyses. In the Random Context condition, we preserved the same shared segments and randomly sampled 30 context segments (each longer than 10 words) from other parts of the novel. For the CLSTM timescale analysis, we used the same 77 long sentences in the Intact Context condition, and randomly sampled 25 context segments (with length > 33 characters) for the Random Context condition.

In brief, the model is processing the same input (the shared segment) with different preceding context (the intact vs. random context). We can now measure the context dependence of individual units by examining how the cell state activations differ between the two conditions, while the network is processing the shared segments with identical input. Any difference in internal representations must arise from the context manipulation, since the current input is the same. Decrease in activation difference over time means that the cell state activation is less and less affected by the context. For a long-timescale unit, whose current state is dependent on information in the far-preceding context, we will see that the activation difference is preserved across contexts (Figure 1B, green curve), even while the unit is processing the shared input. On the other hand, for a short-timescale unit whose activation is driven largely by the current input, we will see that the activation difference drops quickly (Figure 1B, red curve) as the unit processes the shared input.

4 Hierarchical Organization of Timescales Across Layers

Do higher levels of the LSTM model exhibit greater context-dependence? Lakretz et al. (2019) observed that long-range functional units were more common in higher layers, and in general, higher-levels of hierarchical language model exhibit longer range context-dependence (Jain et al., 2019; Jain & Huth, 2018). Therefore, to validate our stimuli and the sensitivity of our methods, we first compared the processing timescales of different hidden layers in both of the LSTMs, by correlating the cell state vectors, column by column, between the Intact condition and Random condition.

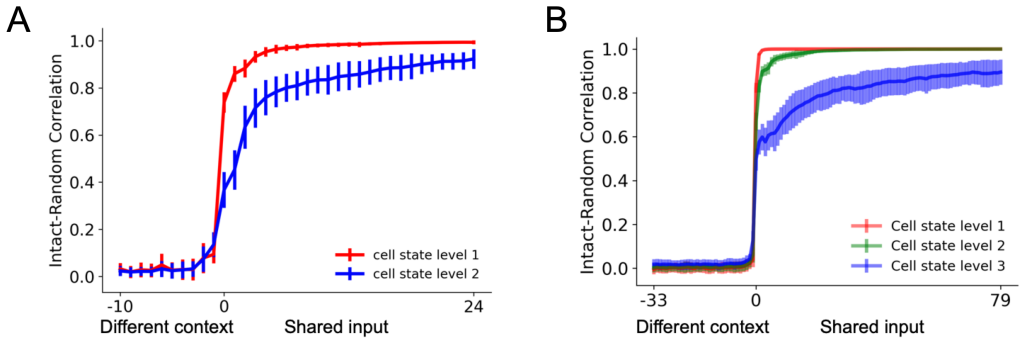

Figure 2: Context effect measured by cell-state vector correlation at different layers in word-level LSTM (WLSTM) and character-level LSTM (CLSTM). **A.** Correlation curves of the WLSTM cell-state vectors across the Intact Context condition and Random Context condition as a function of input token. The correlation is near 0 when the model is processing different context segments, and increase towards 1 when the model starts to process the shared segment. Higher-level cell state showed a more gradual increase in correlation, compared to lower-level cell states, indicating that the higher-levels are more sensitive to the prior context. **B.** As for A, but applied to the three levels of CLSTM. Similar to the WLSTM, higher-level cell state of the CLSTM showed more context sensitivity than the lower-level cell state.

We found that both layers showed near-zero correlation when processing the different context, and the correlation increased as they began to process the shared input. In the WLSTM, the correlation increased more slowly for second-level cell state vectors than for first-level cell state vectors. Thus, the representation of second-level cell state is more sensitive to the different context than the first level. Similarly, for the CLSTM model, the third-level cell state exhibited longer-lasting context sensitivity than lower levels (Figure 2). This observation of longer context-dependence in higher stages of processing is consistent with prior machine learning analyses (Lakretz et al., 2019; Jain & Huth, 2018) and is also analogous to what is seen in the human brain (Hasson et al., 2015; Chien & Honey, 2020; Lerner et al., 2011; Jain et al., 2019). Based on the finding of longer context dependence in higher layers, we examined single units in the highest level hidden units, i.e. the second level of WLSTM (n=650) and the third level of CLSTM (n=1024).

5 Processing Timescales of Individual Units Within LSTM Layers

5.1 Quantifying Single Unit Timescales

We examined the absolute single unit activation difference when processing the shared segments preceded by different context. As expected, most of the hidden units showed different activation when the input tokens were different (i.e. while processing the non-shared context in the Intact Context and Random Context conditions). However, once the shared input tokens begin (at $t = 0$) the Intact-Random activation differences drop (Figure A.1A, A.1B).

We used the rate at which the curves drop to quantify the processing timescale, as this is a measure of how quickly the responses align across different context conditions. To quantify the timescale of individual units, we fit the activation difference curves with a logistic function:

$$Y(x) = \frac{L}{1 + e^{-k(x - x_0)}} + d \tag{1}$$

As shown in Figure A.1A and Figure A.1B, the logistic function fit the raw activation difference curves. We then computed the "timescale" of each unit as the time-to-half-maximum of the logistic decay. In particular, for the WLSTM we used the activation difference $Y(0)$ at the beginning of the shared segment, and at the end of the shared segment $Y(24)$ ($Y(79)$ for the CSLTM) to calculate

the time-to-half-maximum of unit i as:

$$\text{timescale}_i = \lceil Y^{-1}(\frac{Y_i(0) - Y_i(24)}{2}) \rceil \tag{2}$$

where the inverse function $Y^{-1}(y)$ identifies the largest integer t, for which $Y(t) < y$. We included 635 units in WLSTM and 1012 units in CLSTM for further analysis after excluding the units which could not be accurately fit by a logistic function (See Appendix A.1).

5.2 Distribution of Processing Timescales in a Word-level LSTM

The results showed that of the 635 WLSTM units whose processing timescale we mapped, approximately 70% of the units were insensitive to long-range context (processing timescale < 3 words): their activation difference dropped immediately at onset of the shared segment. In contrast, only approximately 13% of the units had a timescales > 7 words (Figure A.2A). Figure 3A shows the absolute activation difference of all units in WLSTM sorted by timescale (long to short). Some of the longer-timescale units continued to exhibit a large activation difference even when processing the shared segments for more than 20 tokens. This suggests that some of the units have timescales longer than we measured here, consistent with previous findings that LSTM overall perplexity can be affected by up to 50 ordered context tokens (Khandelwal et al., 2018).

As we were testing the same word-level LSTM previously studied by Lakretz et al. (2019), we began by examining the timescales of hidden-state units that were already known to be involved in processing context-dependence language information: a "singular number unit" 988, a "plural number unit" 776, and a "syntax unit" 1150. We found that, compared to other units, both "number" units had medium timescales (~ 3 words, ranked 129/650 units), while the "syntax" unit had a long timescale (~ 7 words, ranked 64/650 units) (Figure A.1).

5.3 Timescale Variance across Different Datasets and Context Conditions

To ensure that the timescales we measured are robust across datasets, we conducted the same analysis using the Wikipedia testing dataset used in Gulordava et al. (2018). The mapped timescales were highly correlated (r=0.82, p<0.001) across the *Anna Karenina* dataset and the Wikipedia dataset (Appendix A.2.1, Figure A.4A).

Similarly, to confirm that the timescales measured are not specific to our testing using the ", and" conjunction point, we also measured timescales at an alternative segmentation point, and found the timescales largely preserved (r=0.83, p<0.001), except for a subset of units (Appendix A.2.2, Figure A.4B).

Although the timescales of context dependence we measured were based on the "token distance", they may vary under different context conditions, following the "syntactic distance". For example, one would expect that the units should "reset" at the beginning of a new sentence, and show little context dependence under that condition. Indeed, we found that most units showed little context dependence when the context segments ended with a "full stop", which served as a clear signal for the end of a sentence (Appendix A.2.3, Figure A.4C).

Finally, we examined the role of individual words in shaping context representation and affecting the timescales measurement. We did this by shuffling the words in the Random Context condition instead of replacing it with whole segments from other parts of the corpus. We found that while individual words play an important role in shaping the context dependence of unit timescale, several units showed a longer timescale when that prior context is composed of coherently structured language (Appendix A.2.4, Figure A.4D).

5.4 Distribution of Processing Timescales in a Character-level LSTM

We repeated the timescale mapping in the CLSTM model, and again identified a small subset of long-timescale units (Figure 3B, Figure A.2B). Although there are overall more units in CLSTM, over 63% of the units are insensitive to the context (timescale < 3 characters). Fewer than 15% of the units exhibited timescale > 10 characters, and the unit with the longest timescale only dropped to its half-maximum activation-difference after 50 characters of shared input.

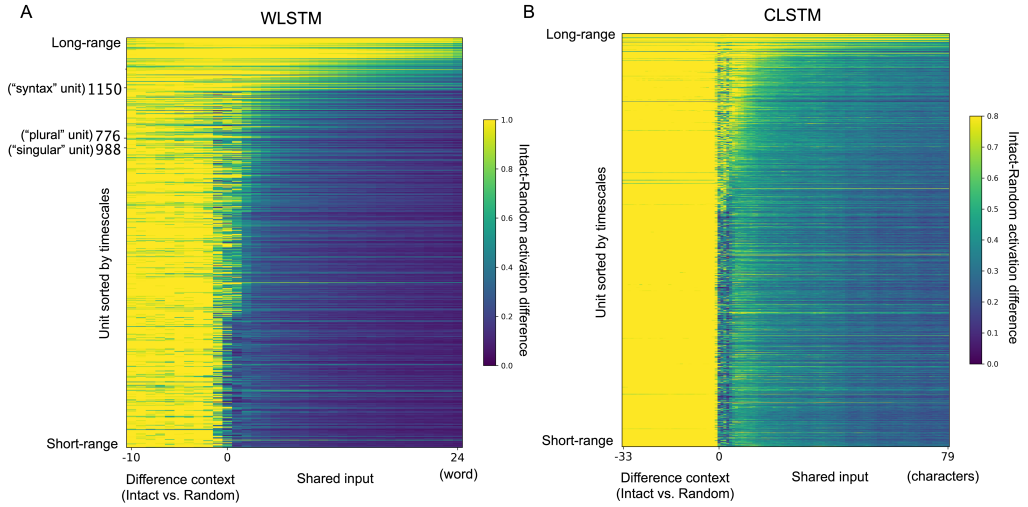

Figure 3: Timescale organization in word-level LSTM (WLSTM) and character-level LSTM (CLSTM) language model. **A.** Absolute activation difference for each WLSTM hidden unit over time, with units (rows) sorted by timescales. A small set of long-timescale units (top) sustain an activation difference during shared segment processing, but most (bottom) are context-insensitive short-timescale units. **B.** Absolute activation difference for each CLSTM unit over time, with units sorted by timescales. Similar to the WLSTM, a small set of long-timescale CLSTM hidden units maintain long-range contextual information.

6 Connectivity of Units with Long and Medium Processing Timescales

Having mapped the timescales of each processing units, we next asked: Does the processing timescale relate to the functional role of each unit in the network? More specifically, are units with longer timescales also units with high degree in the connectivity network? To answer these questions, we analyzed (1) the projection strength of each unit and (2) the similarity of the overall projection pattern (hidden-to-gates) across different units, using the direct weight projections from one hidden unit at time t to the input and forget gate of other hidden units at time $t+1$.

In LSTMs, the amount of contextual (c_{t-1}) and input ($\tilde{c}_t$) information stored in the cell state (c_t) is determined by the forget gate (f_t) and input gate (i_t) activation (Eq. 3); and the activation of the gates i_t and f_t are determined by the current input at time t and the hidden units at time $t-1$ through weight matrices U and W (Eq. 4, 5).

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{3}$$

$$i_t = \sigma(U_i x_t + W_i h_{t-1} + b_i) \tag{4}$$

$$f_t = \sigma(U_f x_t + W_f h_{t-1} + b_f) \tag{5}$$

Here, we are interested in understanding how the contextual information over different timescales is projected from the hidden units to the input and forget gates of other units, and further influence the update of cell states. Thus, we analyzed the network connectivity focusing on the weight matrices W_i and W_f within the highest layer of the WLSTM or CLSTM.

6.1 Strong Projections from Long-Timescale Hidden Units to Gate Units

Units with longer processing timescales made a larger number of strong projections (|z-score|>5, Appendix A.3) to the input and forget gates of other units in both WLSTM (r=0.31, p<0.001, Figure 4A) and CLSTM models (r=0.24, p<0.001, Figure A.5A). Further, we found that the "syntax" unit (Unit 1150) reported by Lakretz et al. (2019) in WLSTM possessed the largest number of strong projections to the input and forget gates of all other units, and the target units that Unit 1150 strongly project to are units with medium- to long-timescale units (Figure 4B).

6.2 CONTROLLER UNITS IN A WORD-LEVEL LSTM AND A CHARACTER-LEVEL LSTM

The presence of strong projections from the "syntax" unit to other long-timescale units motivated us to further explore whether high-degree, long-timescale units in the LSTM also densely interconnect to form a "core network", as seen in the brain (Hagmann et al., 2008; Mesulam, 1998; Baria et al., 2013). If so, this set of units may have an especially important role in determining how prior context is updated and how it is used to gate current processing, analogous to the controller system in the brain (Gu et al., 2015). To identify such "controller units", we binarized the network by identifying the top 258 projection weights from the weight matrices (see Appendix A.3), which provided the edges for a network analysis. We then used k-core analysis (Batagelj & Zaversnik, 2003) to identify the "main network core" (the core with the largest degree) of the network (Figure A.3). At the maximal $k = 5$, the k-core analysis yielded a set of densely interconnected nodes, composed of many long-timescale and medium-timescale units (Figure A.3), also labeled in red in Figure 4A). We (tentatively) refer to this set as the "controller" set of the network.

We performed the same k-core analyses on the CLSTM and observed similar results that the main core network was composed of disproportionately many medium and long-timescale "controller" units (Figure A.5A).

6.3 PROJECTION SIMILARITY OF CONTROLLER AND INTEGRATOR UNITS

We used multi-dimensional scaling (MDS) to visualize the similarity of projection patterns across LSTM units. We recovered a 2-dimensional MDS embedding, where the inter-unit distances was defined based on the similarity of their hidden-to-gate projection patterns (i.e., values in the unthresholded LSTM weight matrices W_i and W_f). We visualized the MDS solution as a graph structure, in which each node is a unit, and the edges reflect connection properties of that unit. Figure 4D shows the resulting 2-D space, with units color-coded by their timescale. The size of the nodes indicates the number of strong projections from the node, and the lines indicate those projections (to forget gate and input gate, based on the color of the line).

"Controller units" (labeled on Figure 4D) were positioned around the periphery of the MDS space, suggesting that these units have distinct projection patterns, both from other "controller" units and from the rest of the network. In contrast, we observed several long-timescale units positioned in the center of the MDS space, suggesting that the projection patterns of these units are close to the mean projection pattern. We refer to this more MDS-central set as the "integrator units" (labeled in green in Figure 4A).

Similar to the WLSTM, the projection patterns of the "controller units" in CLSTM were distinct from other units in the network, according to the MDS results (Figure A.5C). However, we did not observe the "integrator units" positioned in the center of the MDS space of the CLSTM as was observed in the WLSTM.

6.4 ROLES OF CONTROLLER AND INTEGRATOR UNITS

Are the "controller" and "integrator" units important for the model's ability to predict the next token? To test the functional importance of these subsets of units, we conducted group ablation analyses (See Appendix A.5). Ablating controller units reduced the accuracy of token prediction overall, while ablating integrator units only reduced prediction accuracy for the last words of the sentences (Figure 4C). The results confirm that the putative controller and integrator nodes are functionally significant, with distinctive roles in the WLSTM language model.

Finally, to test the generalization of the timescale and connectivity analyses to a different model architecture, we conducted a preliminary analysis of a Gated Recurrent Unit (GRU) language model (Cho et al., 2014). These GRU results are preliminary, because the model was not fully trained, and its perplexity was an order of magnitude inferior to the LSTM models. We found similar sparsity of long-timescale units in the GRU, but did not observe the same relationship between timescales and connectivity (Appendix A.4; Figure A.6; A.7).

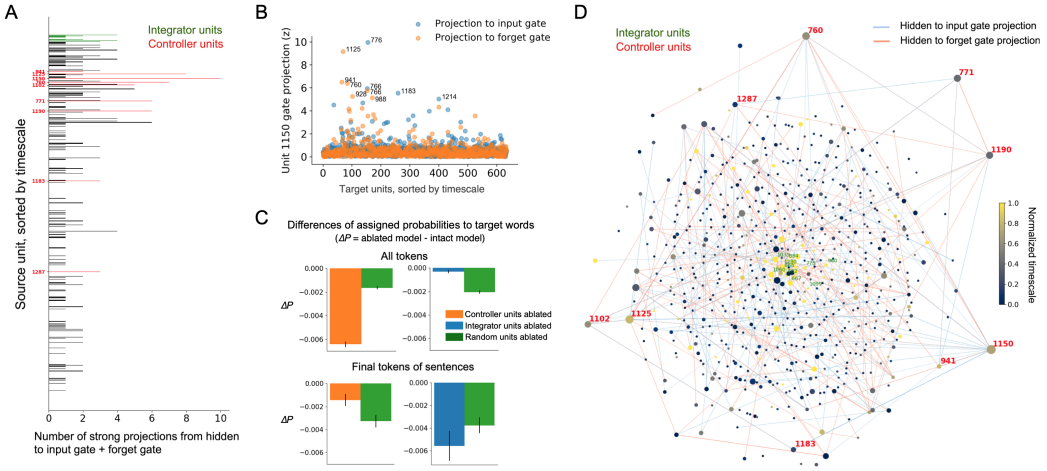

Figure 4: Timescale and connectivity organization in a word-level LSTM. **A.** Long-timescale units show stronger projections from hidden state at time t to the forget gate and input gate at time $t+1$. **B.** Strength of hidden-to-forget gate projections and hidden-input gate projections from high-degree Unit 1150 (a "syntax" unit), to all other units. The units receiving strong projections ($|$z-score$|> 5$) are labeled. **C.** Directed graph of the main core network formed by "controller" units in the WLSTM. **D.** Multi-dimensional scaling representation of network connectivity, with overlaid timescale and strong-connection information. The distance between two nodes indicates the similarity of their hidden-to-gate connection patterns. The size of each node indicates its degree (the number of strong projections from that node to the gate units). An edge between nodes indicates a significant projection between them (i.e. from one node to the gate controlling the other node).

7 Discussion

We demonstrated a new method for mapping the timescale organization in recurrent neural language models. Using this method, we mapped the timescale distributions of units within word-level and character-level LSTM language models, and identified a small set of units with long timescales. We then used network analyses to understand the relationship between the timescale of a unit and its connectivity profile, and we distinguished two subsets of long-timescale units with seemingly different functions. Altogether, we proposed methods combining timescale and connectivity analyses for discovering timescale and functional organization in language models.

The units with long processing timescales included some units whose role in long-range language dependencies had already been established (Lakretz et al., 2019), but almost all of the long timescale units are of unknown function. The timescale mapping procedure described here provides a model-free method for identifying nodes necessary for long-range linguistic and discursive processes (e.g. tracking whether a series of words constitutes an assertion or a question). Future studies of these neural language models could focus on the functions of long-timescale units, and especially the "controller" units, to understand why each unit is tracking long-range information, and how they control the information flow of other units in the network. Moreover, it will be important to investigate whether the processing timescales characterized via token distance (as demonstrated in the current study) are comparable to those measured using functional measures, such as syntactic distance. Relatedly, while we explored the timescale variance under several context conditions, a more thorough investigation will be needed to examine how the timescales of individual units may vary at different positions within a sentence, both in terms of token location and syntactic location.

Processing timescales may exhibit an analogous hierarchical organization in LSTMs and in the human cerebral cortex: in both cases, a subset of nodes with high degree and high inter-connectivity express unusually long timescales. More detailed testing of this apparent correspondence is required, however, because units within an LSTM layer are not spatially embedded, and thus may not show a spatially graded timescale topography.

REFERENCES

Christopher Baldassano, Janice Chen, Asieh Zadbood, Jonathan W Pillow, Uri Hasson, and Kenneth A Norman. Discovering event structure in continuous narrative perception and memory. *Neuron*, 95(3):709–721, 2017.

Alex T Baria, A Mansour, Lejian Huang, Marwan N Baliki, Guillermo A Cecchi, M-Marsel Mesulam, and A Vania Apkarian. Linking human brain local activity fluctuations to structural and functional network architectures. *Neuroimage*, 73:144–155, 2013.

Vladimir Batagelj and Matjaz Zaversnik. An o (m) algorithm for cores decomposition of networks. *arXiv preprint cs/0310049*, 2003.

Hsiang-Yun Sherry Chien and Christopher J Honey. Constructing and forgetting temporal context in the human cerebral cortex. *Neuron*, 2020.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 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)*, pp. 1724–1734, 2014.

Joaquın M Fuster. Network memory. *Trends in neurosciences*, 20(10):451–459, 1997.

Shi Gu, Fabio Pasqualetti, Matthew Cieslak, Qawi K Telesford, B Yu Alfred, Ari E Kahn, John D Medaglia, Jean M Vettel, Michael B Miller, Scott T Grafton, et al. Controllability of structural brain networks. *Nature communications*, 6(1):1–10, 2015.

Kristina Gulordava, Piotr Bojanowski, Édouard Grave, Tal Linzen, and Marco Baroni. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1195–1205, 2018.

Patric Hagmann, Leila Cammoun, Xavier Gigandet, Reto Meuli, Christopher J Honey, Van J Wedeen, and Olaf Sporns. Mapping the structural core of human cerebral cortex. *PLoS Biol*, 6(7):e159, 2008.

Michael Hahn and Marco Baroni. Tabula nearly rasa: Probing the linguistic knowledge of character-level neural language models trained on unsegmented text. *Transactions of the Association for Computational Linguistics*, 7:467–484, 2019.

U. Hasson, E. Yang, I. Vallines, D. J. Heeger, and N. Rubin. A Hierarchy of Temporal Receptive Windows in Human Cortex. *Journal of Neuroscience*, 28(10):2539–2550, March 2008. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.5487-07.2008. URL `http://www.jneurosci.org/cgi/doi/10.1523/JNEUROSCI.5487-07.2008`.

Uri Hasson, Janice Chen, and Christopher J Honey. Hierarchical process memory: memory as an integral component of information processing. *Trends in cognitive sciences*, 19(6):304–313, 2015.

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

Christopher J Honey, Thomas Thesen, Tobias H Donner, Lauren J Silbert, Chad E Carlson, Orrin Devinsky, Werner K Doyle, Nava Rubin, David J Heeger, and Uri Hasson. Slow cortical dynamics and the accumulation of information over long timescales. *Neuron*, 76(2):423–434, 2012.

Shailee Jain and Alexander Huth. Incorporating Context into Language Encoding Models for fMRI. *Advances in Neural Information Processing Systems*, pp. 6628–6637, 2018.

Shailee Jain, Amanda LeBel, and Alexander Huth. Improving language encoding of fMRI responses with transformers. *Annual Meeting of the Society for Neuroscience.*, 2019.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. Sharp nearby, fuzzy far away: How neural language models use context. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 284–294, 2018.

Yair Lakretz, Germán Kruszewski, Théo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. The emergence of number and syntax units in lstm language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 11–20, 2019.

Yair Lakretz, Stanislas Dehaene, and Jean-Rémi King. What limits our capacity to process nested long-range dependencies in sentence comprehension? *Entropy*, 22(4):446, 2020.

Yulia Lerner, Christopher J Honey, Lauren J Silbert, and Uri Hasson. Topographic mapping of a hierarchy of temporal receptive windows using a narrated story. *Journal of Neuroscience*, 31(8): 2906–2915, 2011.

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

M-Marsel Mesulam. From sensation to cognition. *Brain: a journal of neurology*, 121(6):1013–1052, 1998.

Caroline A Runyan, Eugenio Piasini, Stefano Panzeri, and Christopher D Harvey. Distinct timescales of population coding across cortex. *Nature*, 548(7665):92–96, 2017.

Leo Tolstoy. *Anna karenina*. Lulu. com, 2016.

Jiang Xu, Stefan Kemeny, Grace Park, Carol Frattali, and Allen Braun. Language in context: emergent features of word, sentence, and narrative comprehension. *NeuroImage*, 25(3):1002–1015, April 2005. ISSN 10538119. doi: 10.1016/j.neuroimage.2004.12.013. URL `http://linkinghub.elsevier.com/retrieve/pii/S1053811904007748`.

A Appendix

A.1 Units excluded from timescale analysis

We excluded 1 unit in the WLSTM model and 5 units in CLSTM model which were not properly fit using the logistic function; we further excluded 14 units in the WLSTM model and 7 units in the CLSTM model which either did not show a non-zero activation difference before the shared segment started, or whose activation differences increased when started to process the shared segment. After these exclusions, 635 units remained in the WLSTM and 1012 units remained in the CLSTM for further analysis.

A.2 Timescale analyses across different datasets and context conditions

A.2.1 Wikipedia test dataset

The *Anna Karenina* corpus used in the current study has a different linguistic structure from the Wikipedia corpus on which the WLSTM and CLSTM models were trained. Although we analyzed only the *Anna Karenina* sentences with low perplexity, it wsa important to test the robustness of our results across datasets. Thus, we mapped the timescale of each unit using the Wikipedia test set, as used by Gulordava et al. (2018). Specifically, we sampled 500 long sentences containing “, and” for the Intact Context condition. As before, we generated sentences by preceding the “shared input” segment (after the conjunction) with either the original prior context segment, or a randomly chosen prior context segment. Same as the original analysis, we then replaced the context segment with 30 context segments randomly sampled from other parts of the test set for generating the Random Context condition. The mapped timescales using the Wikipedia test set were highly correlated with the novel corpus, suggesting the robustness of unit timescales (Figure A.4A).

A.2.2 Timescales measured in the middle of a sentence

To examine how the timescales of individual units may vary across different positions in a sentence, we varied the location of the segmentation point. Instead of using the conjunction (“, and”) as the

segmentation point, we chose an arbitrary segmentation point: the 15th token of a long sentence, to separate context segment and shared input segment. In the Random Context condition, we replaced the context segment with the first 15 tokens from other sentences of the corpus. We found that the unit timescales were highly correlated with the condition where we used the conjunction as the segmentation point with several units shift their timescales to either directions (Figure A.4B). This analysis was conducted using Wikipedia test set.

A.2.3 TIMESCALE RESET AT THE BEGINNING OF A SENTENCE

To examine if the timescales of individual units can flexibly reset at the beginning of a sentence, we conducted the same timescale analysis but using a "full stop" as the segmentation point instead of the conjunction ", and". Thus, if the original test string was "The girl kicked the call, and the boy caught it", then the full-stop test string would be "The girl kicked the ball. The boy caught it." In this setting, the context segment and shared input segment in the Intact Context condition are two consecutive sentences. To ensure the temporal dependence between the context segment and shared input segment, we sampled 100 consecutive sentence pairs from the *Anna Karenina* corpus. Note that This is not possible using the Wikipedia test set from Gulordava et al. (2018), which is composed of unrelated sentences. The Random Context condition was generated by replacing the first sentence with randomly sampled sentences from other parts of the novel. We found that when using "full stop" to segment context and shared input, most units in the network showed timescale near 0, indicating near-zero dependence on the linguistic context from the text preceding the full stop (Figure A.4C). This suggests that the units in LSTM tend to "reset" their context representation at the beginning of a sentence.

A.2.4 CONTEXT REPRESENTATION SHAPED BY INDIVIDUAL WORDS

Inspired by the token-shuffling procedure of Khandelwal et al. (2018), we explored whether the context representations of individual units in the LSTM were shaped by individual words, rather than coherent sequences of words. For this analysis, instead of replacing the context with syntactically structured segments from other part of the corpus, we generated the "random context" by shuffling the order of words within the context segment. We then mapped the unit timescales as before, by examining the unit activation difference as a function of the distance from the onset of shared input. Intriguingly, we found that most of the units showed similar timescales across the context-replacement and context-shuffling procedures (Figure A.4D). This suggests that the context representations in LSTMs largely depend on the *presence* of individual words in the context, rather than their appearance within coherent linguistic sequences. However, we did observe a subset of units (labeled in the Figure, and almost all long-timescale units) whose timescales were longer when context was replaced rather than shuffled. For this subset of units, the ability to maintain a representation of prior context over many tokens depends on that prior context being a coherent linguistic sequence. This subset of units are a promising target for future studies of syntactic representations in LSTMs.

A.3 IDENTIFYING STRONG HIDDEN-TO-GATE PROJECTIONS

First, for each hidden unit, we concatenated the corresponding rows in the W_{hi} and W_{hf} matrices, to generate a single "hidden-to-gate" projection vector for that hidden unit. Next we we z-scored the vector to get standardized projection values from that unit to all other units in the network. Using |z-score|$>$ 5 as criterion, we identified a total of 258 "strong projections" from all hidden units to the input gate and forget gate in the WLSTM. The projection strength of each unit was then calculated based on its number of "strong projections" (Figure 4A). Although the criterion |z-score|$>$ was selected to better visualize the results in Figure 4, different criteria did not change the results that units with longer timescales have more strong projections. For example, using |z-score|$>$ 3 as threshold we obtained corr(timescale, projections) = 0.30, p$<$0.001; |z-score|$>$ 4 we obtained corr(timescale, projections) = 0.35, p$<$0.001.

Next, we identified the edges corresponding to the top 258 magnitude weight-values within the combined W_{hi} and W_{hf} matrices. Together, these edges formed a "strong-projection network". Finally, we used k-core analysis to identify the main core of the strong-projection network. This main core composed our "controller units" (Figure A.3).

Using the same criteria and method, we identified a total of 390 "strong projections" from all hidden units to the input gate and forget gate in the CLSTM. We then extracted the top 390 weight values from the weight matrices to construct a "strong-projection network" and again identified the main core network, composed the "controller units" for the CLSTM model (Figure A.5A, A.5B)

A.4 Mapping timescale organization in a GRU language model

A.4.1 Training

To explore whether the timescale mapping methods, and our findsings, may generalize to other model architectures, we trained and studied a word-level GRU language model (Cho et al., 2014). As far as possible, we applied the same parameters in the GRU as were used for the LSTM by Gulordava et al. (2018): the same Wikipedia training corpus, the same loss function (i.e. cross-entropy loss), and the same hyperparameters. Thus, the GRU model also had two layers, with 650 hidden units in each layer.

Due to limitations of time and computational resources, we had to stop training the GRU model after 48 hours (10.5 epochs), at which point the GRU achieved a test perplexity of 349.39. This perplexity is much higher than that of the LSTM model reported in Gulordava et al. (2018) (perplexity = 52.1 in the English corpora, after training for 40 epochs and selecting the model with the lowest perplexity out of 68 combinations of different hyperparamters). In spite of the fact that the GRU model was sub-optimal, we analyzed the timescale of its hidden units using the same method as was used for analyzing the LSTMs, and using the test data derived from the training Wikipedia corpus.

A.4.2 Timescale organization of a GRU model

Similar to the LSTM model of Gulordova et al, the majority of the units in the GRU also showed shorter timescales. More specifically, we found: (1) the second layer of the GRU model was more sensitive to prior context than the first layer, as in the LSTM (A.6A); (2) the distribution of timescales across units was similar in the GRU and LSTM, although the GRU showed a more right-skewed distribution with a larger proportion of short-timescale units (A.6B, C).

A.4.3 Timescale versus network connectivity in a GRU model

We also performed the timescale vs. network connectivity analyses on the GRU model. Because the update of hidden states in GRU are controlled by the update gate, we measured the projection patterns of hidden units by analyzing the matrix of hidden-to-update-gate weights. In contrast to the LSTM models, hidden units in the GRU that we trained did not show a relationship between longer timescales and stronger hidden-to-gate projections (A.7A). Moreover, when using k-core analysis to identify subunits of interconnected high-degree units, the core network in the GRU contained many units with long to short timescales. Interestingly, when we visualized the position of the k-core units in the MDS space, they tended to locate at the edge of the space, similar to what we found in LSTM. This indicates that, as in the LSTM, the core units in the GRU have distinctive profiles, distant from one another and from other units in the network (A.7B).

These apparent similarities and differences between LSTM and GRU are intriguing, but we emphasize that (1) the perplexity of this GRU model is much higher than the LSTM, due to limitations in training time and tuning, and that (2) comparing the LSTM and GRU connection patterns is not straightforward, as the overall distribution of weights is different. Further work will be required to determine comparable thresholds for "strong" projections and "high-degree units" in each case. As we noted in the manuscript and above, the connectivity results are exploratory; however, we believe that the GRU analysis demonstrates how these methods can be extended to map and compare the functional organization of language models of different architectures.

A.5 Ablation analyses on putative controller and integrator units

To examine the non-trivial roles of the controller and integrator units identified in the word-level LSTM model, we performed a preliminary group ablation analysis to look at how ablating the controller units influences model performance on predicting the next token, relative to the ablation of a random set of units. Specifically, since long-timescale integrator units should have most effect pre-

dicting tokens at the later part of the sentences (i.e., when more context is integrated), we examined the model performance on predicting tokens at two different positions: (1) all the tokens regardless of their positions in the sentences ("All tokens" condition), and (2) the last tokens of sentences ("Final tokens" condition).

We evaluated the effects of ablation on model performance by measuring the differences of probabilities (ΔP) assigned to the target words (ΔP = probability of target word in ablated model minus probability of target word in original model). Ablation effects for controller units (N=9) and integrator units (N=10) were compared against a baseline of ablating the same number of randomly-selected units from layer 2 of the LSTM (Figure 4C). We used the test corpus used by Gulordava et al. (2018) and measured the average performance of each model across 100 text-batches, randomly sampled from the Wikipedia test dataset. Each text-batch was composed of 1000 tokens that start at the beginning of a sentence.

In the "All tokens" condition, we calculated the ΔP for every token in the tested text, while in the "Final tokens" condition, we calculated ΔP only at the last token of every sentence (i.e. the token right before the full stop"." of each sentence). We then average the ΔP in both conditions across text-batches to get a mean performance difference between the ablated model and the intact model.

Ablating controller units reduced the probabilities assigned to the target words, more so than ablating random units (Figure 4C, controller vs. random across 100 text batches: Cohen's d = -4.85, t = -34.28, p$<$0.001). In contrast, ablating integrator units showed reduced the probabilities less than ablating random units (integrator vs. random: Cohen's d = 2.50, t = 17.67, p$<$0.001). We hypothesized that that the integrator units mostly influence the model performance on predicting tokens in cases where long-range information is especially relevant, such as in the later portions of clauses and sentences. Consistent with this, we found that, when we examined the ablation effects only for tokens in the final position of a sentence, ablating integrator units reduced the probabilities more than ablating random units (Cohen's d = -0.34, t = -2.41, p = 0.017). Interestingly, ablating controller units reduced the probability of sentence-final targets less than random units (Cohen's d = 0.67, t = 4.74, p$<$0.001).

In summary, these ablation results indicate a non-trivial functional role for the controller and integrator units, despite the fact that each subset of units is composed of only 10 amongst 650 total hidden units. Also, the putative controller and integrator sets appear to have distinctive roles within the WLSTM, with the controllers supporting accurate predictions overall, while the integrator units appear to boost accurate predictions at the end of sentences.

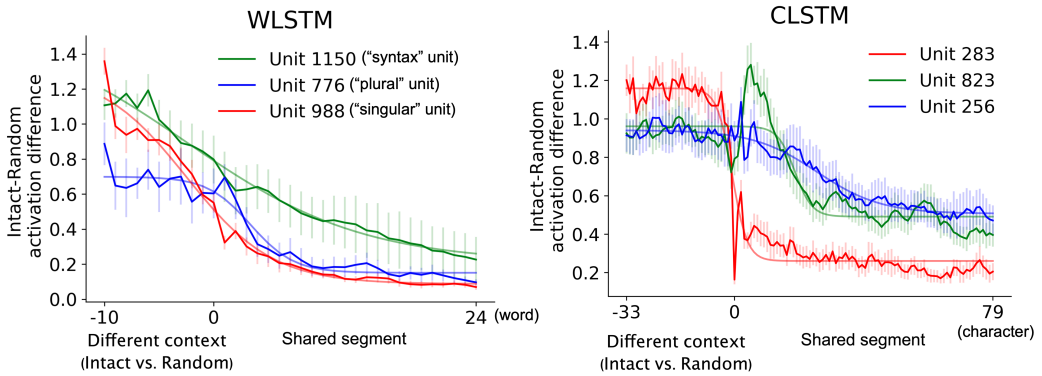

Figure A.1: Example single units activation differences across tokens in word-level LSTM (WLSTM) and character-level LSTM (CLSTM) language models. **A.** Single units activation differences across tokens in the WLSTM, with logistic curve fits overlaid. The error bars indicate 95% confidence interval across trials. The example units include the functional units reported by (Lakretz et al., 2019). **B.** Example units activation differences in CLSTM model. The three units were randomly selected from long-, medium- and short-timescales units.

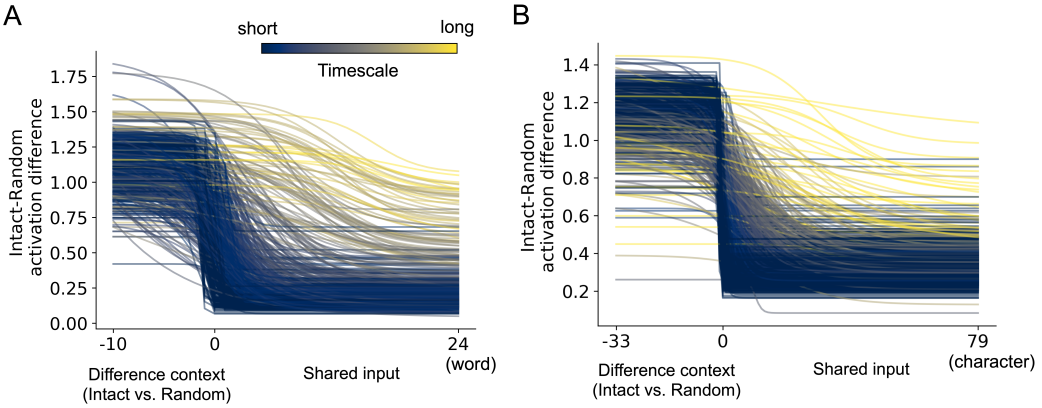

Figure A.2: Logistic fitted curves of activation difference in individual units in word-level and character-level LSTM language model, colored by timescale. **A.** Logistic fitted curves of activation difference over time of all units in word-level LSTM. The color indicates the timescale measured by full-width half-maximum (FWHM) of the curve. **B.** Logistic fitted curves of activation difference over time of all units in character-level LSTM.

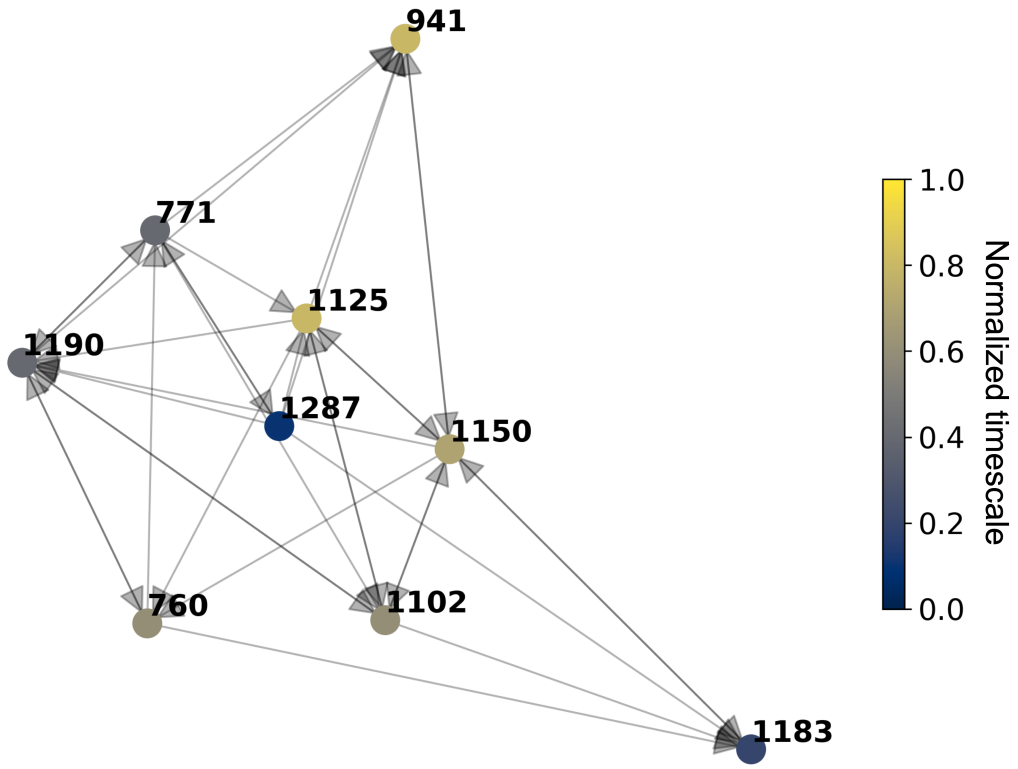

Figure A.3: The core network consists of “controller units” identified in word-level LSTM model.

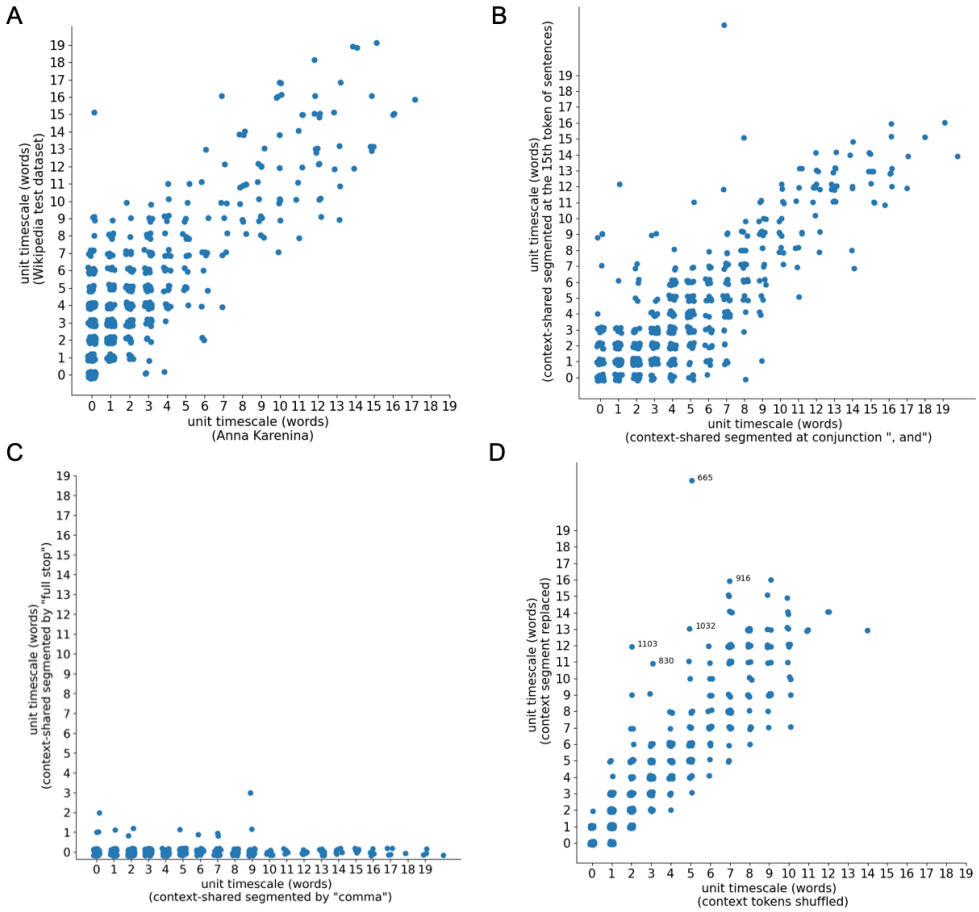

Figure A.4: Mapped timescales across different datasets and context conditions. **A.** Timescales measured using Wikipedia test set from Gulordava et al. (2018) are highly correlated with the timescales measured using *Anna Karenina* (r=0.82, p<0.001). **B.** Timescales measured between sentences (i.e., using "full stop" to segment context and shared segments), vs. within sentence (i.e., using "comma" to segment context and shared segments). Most of the units showed little context dependence when the segmentation performed at the beginning of a sentence, suggesting a "reset" of context representation in these units. **C.** Timescales measured in the middle of a sentence (based on the token number) are highly correlated with timescales measured at the conjunction of a sentence(", and") (r=0.83, p<0.001). **D.** Timescales measured by replacing the context with syntactically structured segments vs. with the shuffled context tokens. Most of the units showed similar timescales under the two conditions, and several units showed longer timescales (i.e. preserved more context) when the context segments were syntactically structured.

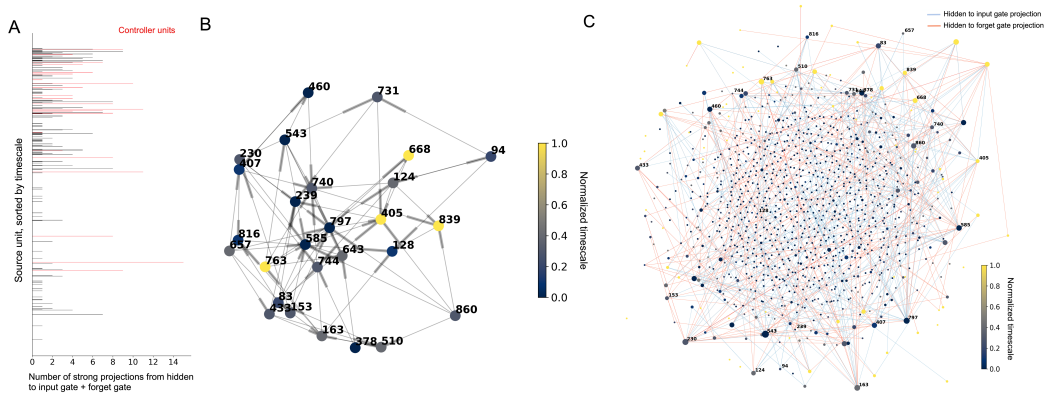

Figure A.5: Timescale and connectivity organization in a character-level LSTM (CLSTM). **A.** Longer-timescale units have overall stronger projections from the hidden units at time t to the forget gates and input gates at time $t+1$. **B.** The main core network ($k = 4$) formed by the controller units in the CLSTM. **C.** The multidimensional scaling space of the hidden-to-gate connection pattern of all units. The distance between nodes indicates their hidden-to-gate connection similarity; the size of the node indicates the number of strong projections from the node; and the line between two nodes indicates a significant projections between them.

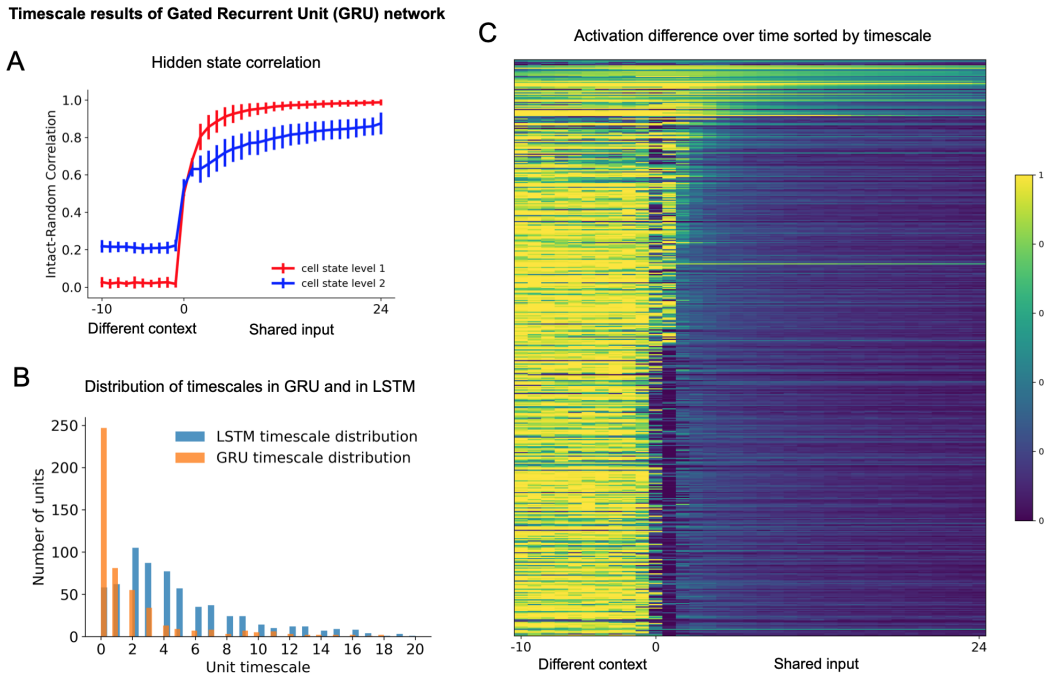

Figure A.6: Mapping timescale organization in GRU language model. **A.** similar to the word level LSTM, the second layer of the GRU model was more sensitive to prior context than the first layer. **B.** Distributions of unit timescales in GRU and in LSTM. While both models showed sparse long-timescale units compared to rich short-timescale units, the GRU showed a right-skewed distribution with a larger proportion of short-timescale units relative to the word-level LSTM. **C.** Absolute activation difference for each second-layer GRU unit over time.

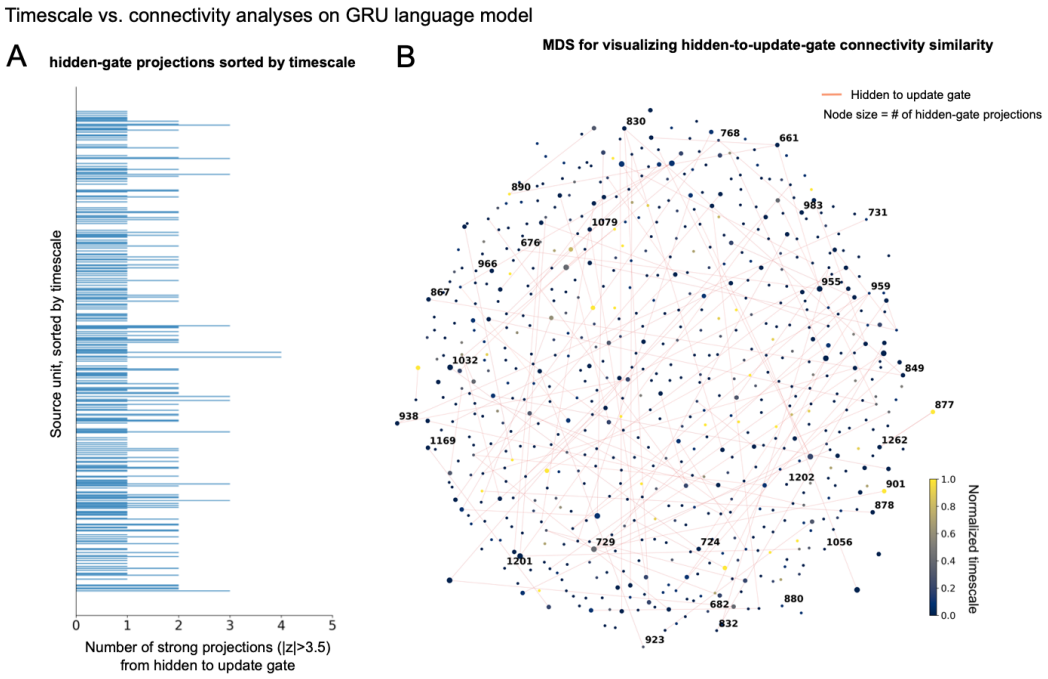

Figure A.7: Timescale-connectivity analyses in GRU language model. **A.** Different from LSTM, the GRU we analyzed did not show the pattern that units with longer timescales exhibited more strong projections. **B.** Multi-dimensional scaling (MDS) results in GRU. The "controller units" identified in GRU using k-core analysis (labeled on the graph) tend to locate at the edge of the MDS space, similar to the LSTM.