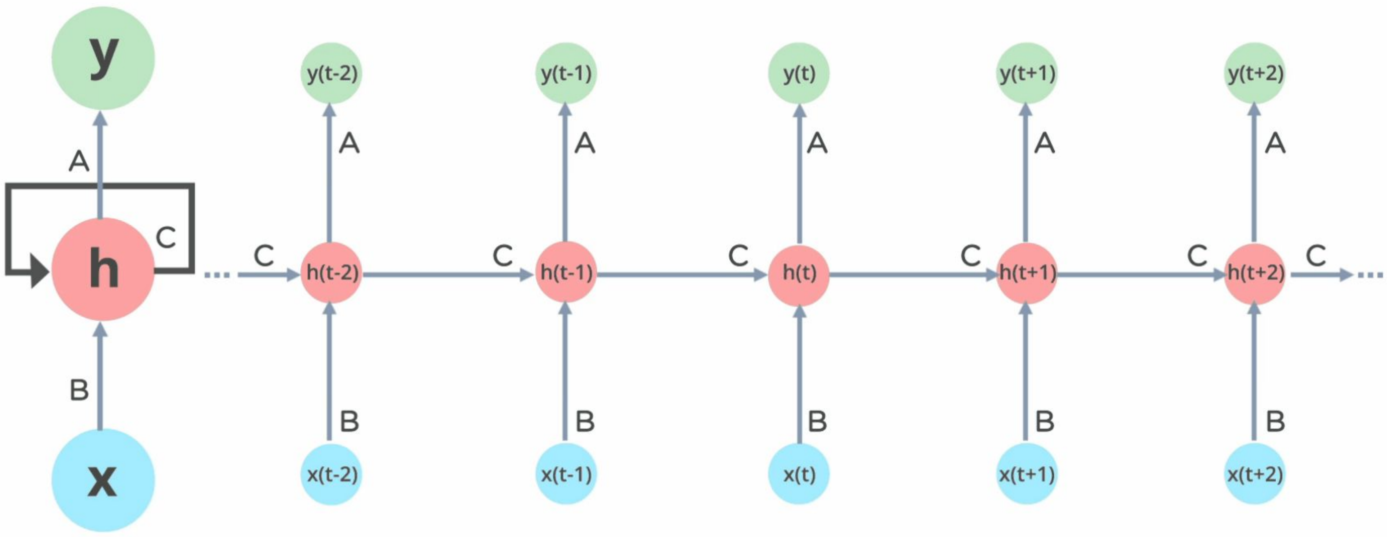
Understanding Hidden Memories of Recurrent Neural Networks

Yao Ming, Shaozu Cao, Ruixiang Zhang, Zhen Li, Yuanzhe Chen, Yangqiu Song, and Huamin Qu

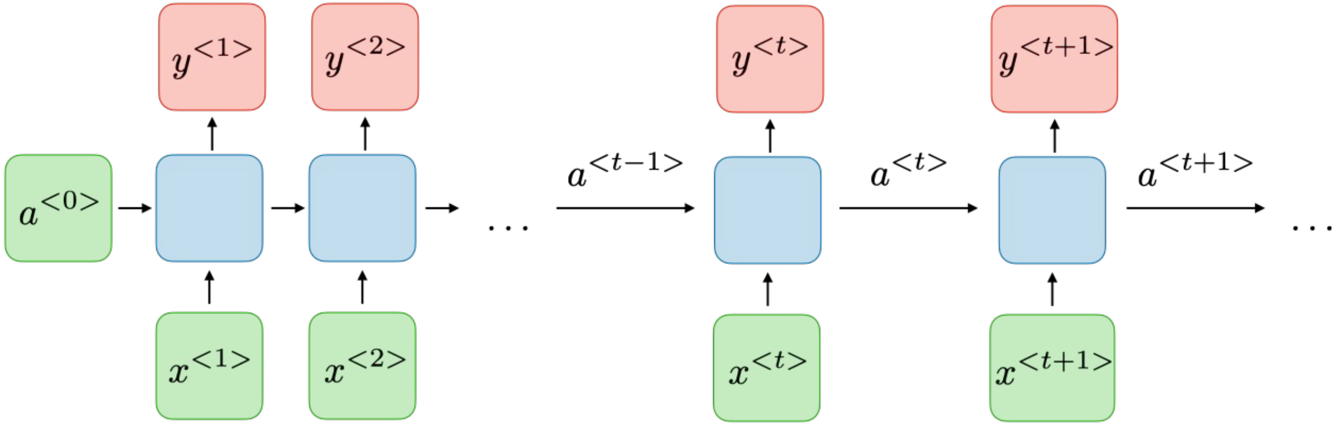
Main Concepts

What is Recurrent Neural Networks (RNN)?



- RNN use the same weights for each element of the sequence.
- Decreasing the number of parameters.
- Allows the model to generalize to sequences of varying lengths.
- A RNN can anticipate sequential data in a way that other algorithms can't.

The Architecture of a Traditional RNN



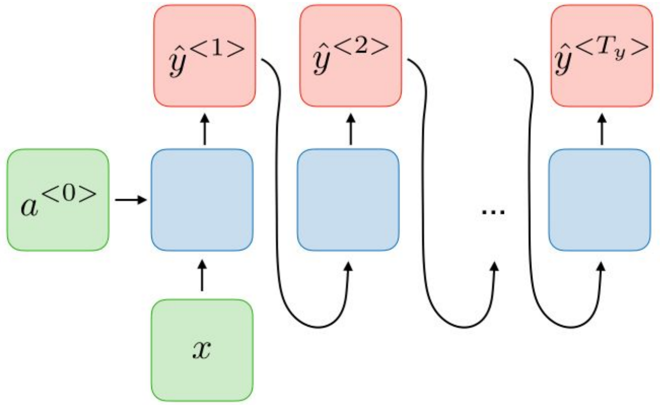
For each timestep t, the activation $a^{<t>}$ and the output $y^{<t>}$ are expressed as follows:

$$\boxed{a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)} \quad \text{and} \quad \boxed{y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)}$$

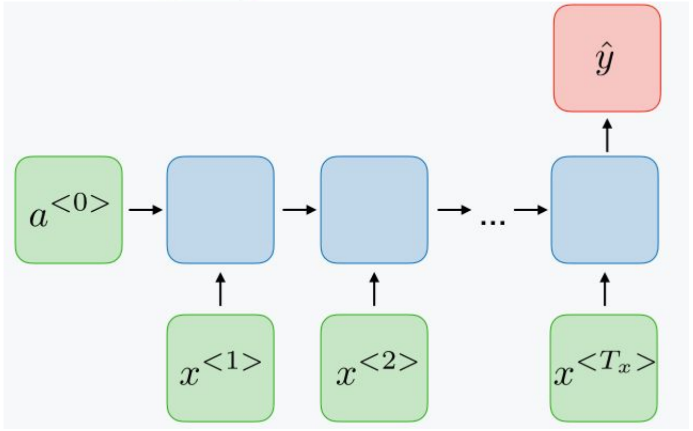
where $W_{ax}, W_{aa}, W_{ya}, b_a, b_y$ are coefficients that are shared temporally and g_1, g_2 activation functions.

Types of RNN

- **One to Many**: There is only one pair here. A one-to-one architecture is used in traditional neural networks. E.g, Music generation.

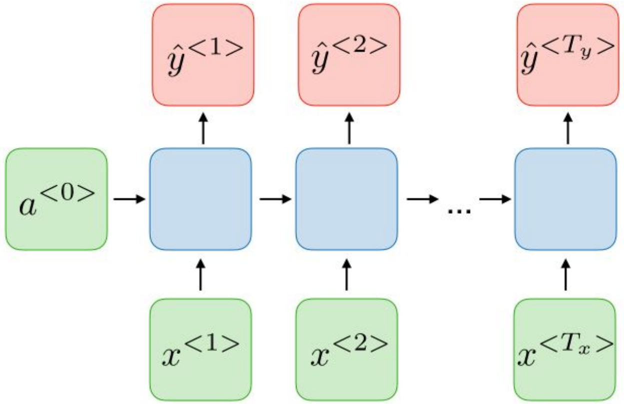


- **Many To One:** A single output is produced by combining many inputs from distinct time steps. E.g., Sentiment analysis and emotion identification

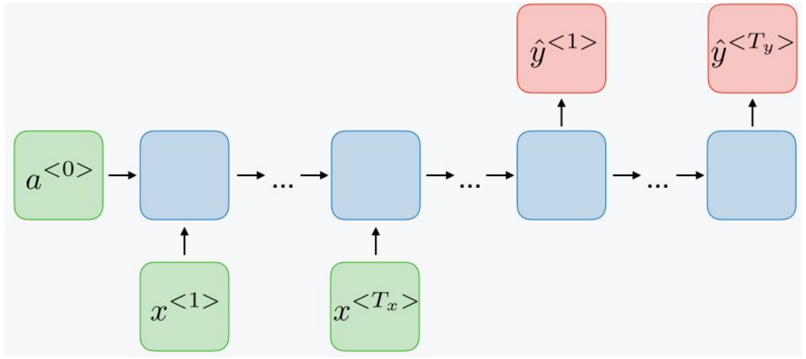


Types of RNN

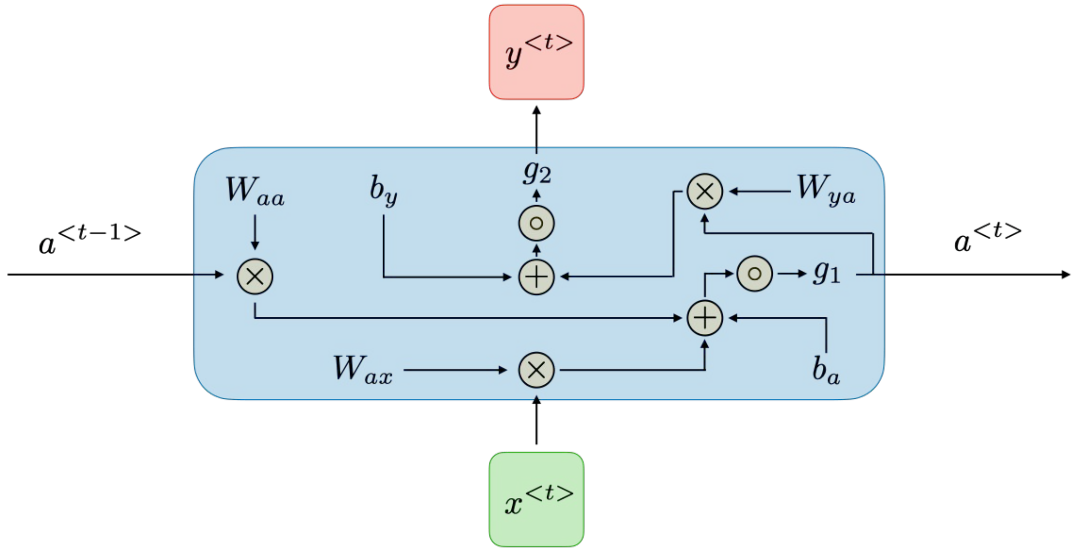
- **Many to Many**: Each single input has an output. e.g., Language modeling.



- **Many To Many:** Multiple sequence of outputs from multiple sequence of inputs. e.g., Machine Translation.



Forward propagation

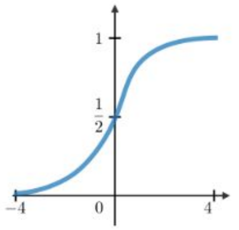
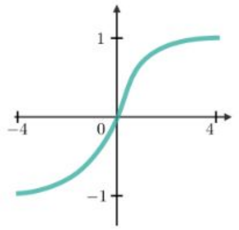
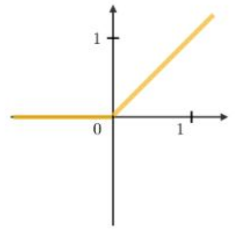


For each time step t, the activation a<t> and the output y<t> is expressed as follows:

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \qquad \hat{y}^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

Forward propagation and Loss Functions

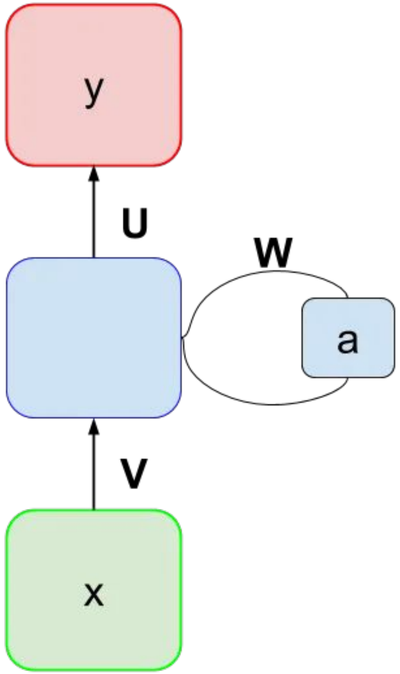
In our model, g1 usually is **Tanh** or **ReLU** and g2 is **sigmoid** or **Softmax** (depends on how variables you do like to identify)

Sigmoid	Tanh	RELU
$g(z) = \frac{1}{1+e^{-z}}$	$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0, z)$
		

In the case of a recurrent neural network, the loss function L of all time steps is defined based on the loss at every time step as follows:

$$L(\hat{y}, y) = \sum_{t=1}^{T_y} E^{(t)} \qquad E^{(t)} = L^{<t>}(\hat{y}^{<t>}, y^{<t>})$$

Backward propagation



We know:

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$

$$\hat{y}^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

Let's define:

$$q^{<t>} = Va^{<t>} + b_y$$

$$z^{<t>} = Wa^{<t-1>} + Ux^{<t>} + b_a$$

We have:

$$a^{<t>} = g_1(z^{<t>})$$

$$\hat{y}^{<t>} = g_2(q^{<t>})$$

Backward propagation

At timestep T, the derivative of the loss L with respect to some weight matrix M is expressed as follows:

$$\frac{\partial L^{(T)}}{\partial M} = \sum_{t=1}^{T} \frac{\partial E^{(T)}}{\partial M}\Big|_{(t)}$$

We can rewrite as (using U, W, V):

$$\frac{\partial L}{\partial U} = \sum_{t=1}^{T_y} \frac{\partial E^{(t)}}{\partial U}\Big|_{(t)} \qquad \frac{\partial L}{\partial W} = \sum_{t=1}^{T_y} \frac{\partial E^{(t)}}{\partial W}\Big|_{(t)} \qquad \frac{\partial L}{\partial V} = \sum_{t=1}^{T_y} \frac{\partial E^{(t)}}{\partial V}\Big|_{(t)}$$

Where:

$$\frac{\partial E^{(t)}}{\partial U} = (\hat{y}^{<t>} - y^{<t>}).\, V.\sum_{k=0}^{t}\Big[\frac{\partial a^{<t>}}{\partial a^{<k>}}\frac{\partial a^{<k>}}{\partial z^{<k>}}.\,(x^{<k>})^T\Big]$$

$$\frac{\partial E^{(t)}}{\partial W} = (\hat{y}^{<t>} - y^{<t>}).\, V.\sum_{k=0}^{t}\Big[\frac{\partial a^{<t>}}{\partial a^{<k>}}\frac{\partial a^{<k>}}{\partial z^{<k>}}.\,(a^{<k-1>})^T\Big]$$

$$\frac{\partial E^{(t)}}{\partial V} = (\hat{y}^{<t>} - y^{<t>}).\,(a^{<t>})^T$$

Vanishing gradient problem

The reason why they happen is that it is difficult to capture long term dependencies

$$\frac{\partial a^{<t>}}{\partial a^{<k>}} = \frac{\partial a^{<t>}}{\partial a^{<t-1>}} \frac{\partial a^{<t-1>}}{\partial a^{<t-2>}} \cdots \frac{\partial a^{<k+2>}}{\partial a^{<k+1>}} \frac{\partial a^{<k+1>}}{\partial a^{<k>}}$$

$$\frac{\partial a^{<t>}}{\partial a^{<k>}} = \prod_{i=k+1}^{t} \frac{\partial a^{<i>}}{\partial a^{<i-1>}} \qquad \frac{\partial a^{<t>}}{\partial a^{<k>}} = \prod_{i=k+1}^{t} W^T diag[\frac{\partial g_1(a^{<i-1>})}{\partial a^{<i-1>}}]$$

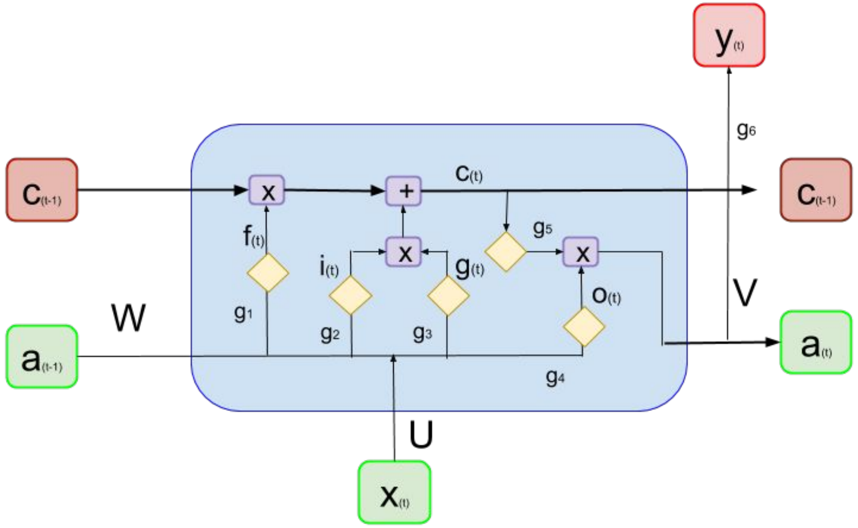
Taking non-linear functions to analyze, we obtain:

$$\|diag[\frac{\partial g_1(a^{<i-1>})}{\partial a^{<i-1>}}]\| \le \gamma \qquad \|\frac{\partial a^{<i>}}{\partial a^{<i-1>}}\| \le \|W^T\| \|diag[\frac{\partial g_1(a^{<i-1>})}{\partial a^{<i-1>}}]\| \le \gamma_w . \gamma$$

$$\|\frac{\partial a^{<t>}}{\partial a^{<k>}}\| \le (\gamma_w . \gamma)^{(t-k)} = (\lambda)^{(t-k)}$$

If lambda << 1, Then Vanishing Gradient. Otherwise, lambda >1, Then Exploding Gradient.

Models of RNN: Long Short Term Memory (LSTM)



$$a^{<t>} = o^{<t>} \circ g_5(c^{<t>})$$

$$\hat{y}^{<t>} = g_6(Va^{<t>} + b_y)$$

Forget gate:

$$f^{<t>} = g_1(W_f a^{<t-1>} + U_f x^{<t>} + b_f)$$

Input gate:

$$i^{<t>} = g_2(W_i a^{<t-1>} + U_i x^{<t>} + b_i)$$

Update gate: Candidate

$$g^{<t>} = g_3(W_c a^{<t-1>} + U_c x^{<t>} + b_c)$$

Update gate: Memory

$$c^{<t>} = f^{<t>} \circ c^{<t-1>} + i^{<t>} \circ g^{<t>}$$

Output gate:

$$o^{<t>} = g_4(W_o a^{<t-1>} + U_o x^{<t>} + b_o)$$

Models of RNN: LSTM backpropagation

$$p^{<t>} = g_5(c^{<t>}) \qquad s^{<t>} = W_o a^{<t-1>} + U_o x^{<t>} + b_o$$

$$\frac{\partial E^{(t)}}{\partial V} = (\hat{y}^{<t>} - y^{<t>}).\,(a^{<t>})^T \quad \frac{\partial L}{\partial V} = \sum_{t=1}^{T_y}[(\hat{y}^{<t>} - y^{<t>}).\,(a^{<t>})^T]$$

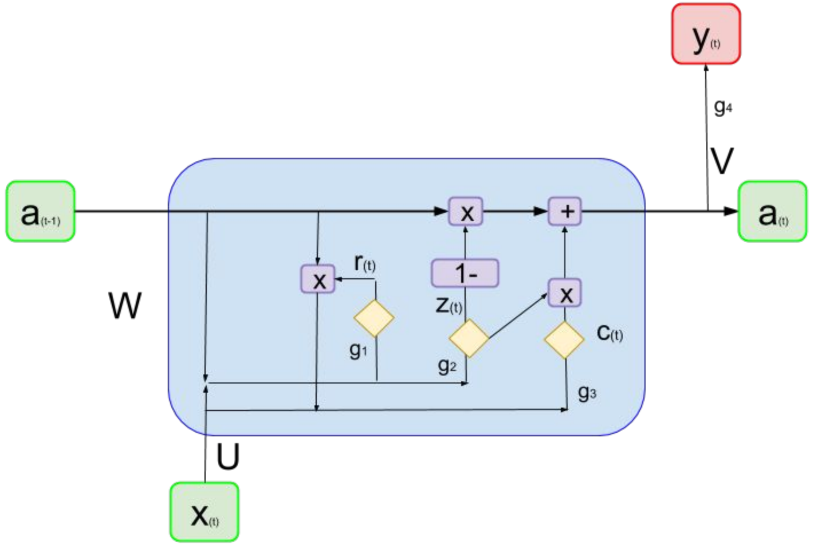
$$\frac{\partial E^{(t)}}{\partial W_o} = (\frac{\partial E^{(t)}}{\partial a^{<t>}} + \frac{\partial E^{(t+1)}}{\partial a^{<t>}}).\,p^{<t>}.\,\frac{\partial o^{<t>}}{\partial s^{<t>}}.\,(a^{<t-1>})^T$$

$$\frac{\partial L}{\partial W_o} = \sum_{t=1}^{T_y}[(\frac{\partial E^{(t)}}{\partial a^{<t>}} + \frac{\partial E^{(t+1)}}{\partial a^{<t>}}).\,p^{<t>}.\,\frac{\partial o^{<t>}}{\partial s^{<t>}}.\,(a^{<t-1>})^T]$$

$$\frac{\partial E^{(t)}}{\partial U_o} = (\frac{\partial E^{(t)}}{\partial a^{<t>}} + \frac{\partial E^{(t+1)}}{\partial a^{<t>}}).\,p^{<t>}.\,\frac{\partial o^{<t>}}{\partial s^{<t>}}.\,(x^{<t>})^T$$

$$\frac{\partial L}{\partial U_o} = \sum_{t=1}^{T_y}[(\frac{\partial E^{(t)}}{\partial a^{<t>}} + \frac{\partial E^{(t+1)}}{\partial a^{<t>}}).\,p^{<t>}.\,\frac{\partial o^{<t>}}{\partial s^{<t>}}.\,(x^{<t>})^T]$$

Models of RNN: Gated Recurrent Unit (GRU)



Update gate:

$$z^{<t>} = g_1(W_z a^{<t-1>} + U_z x^{<t>} + b_z)$$

Reset gate:

$$r^{<t>} = g_2(W_r a^{<t-1>} + U_r x^{<t>} + b_r)$$

Candidate gate:

$$c^{<t>} = g_3(W_c(r^{<t>} \circ a^{<t-1>}) + U_c x^{<t>} + b_c)$$

$$a^{<t>} = (1 - z^{<t>}) \circ a^{<t-1>} + z^{<t>} \circ c^{<t>}$$

$$\hat{y}^{<t>} = g_4(V a^{<t>} + b_y)$$

Models of RNN: GRU backpropagation

$$s^{<t>} = W_c(r^{<t>} \circ a^{<t-1>}) + U_c x^{<t>} + b_c$$

$$\frac{\partial E^{(t)}}{\partial V} = (\hat{y}^{<t>} - y^{<t>}).\,(a^{<t>})^T \qquad \frac{\partial L}{\partial V} = \sum_{t=1}^{T_y}[(\hat{y}^{<t>} - y^{<t>}).\,(a^{<t>})^T]$$

$$\frac{\partial E^{(t)}}{\partial W_c} = (\frac{\partial E^{(t)}}{\partial a^{<t>}} + \frac{\partial E^{(t+1)}}{\partial a^{<t>}}).\,z^{<t>}\frac{\partial c^{<t>}}{\partial s^{<t>}}.\,(r^{<t>} \circ a^{<t-1>})^T$$

$$\frac{\partial L}{\partial W_c} = \sum_{t=1}^{T_y}[(\frac{\partial E^{(t)}}{\partial a^{<t>}} + \frac{\partial E^{(t+1)}}{\partial a^{<t>}}).\,z^{<t>}\frac{\partial c^{<t>}}{\partial s^{<t>}}.\,(r^{<t>} \circ a^{<t-1>})^T]$$

$$\frac{\partial E^{(t)}}{\partial U_c} = (\frac{\partial E^{(t)}}{\partial a^{<t>}} + \frac{\partial E^{(t+1)}}{\partial a^{<t>}}).\,z^{<t>}\frac{\partial c^{<t>}}{\partial s^{<t>}}.\,(x^{<t>})^T$$

$$\frac{\partial L}{\partial U_c} = \sum_{t=1}^{T_y}[(\frac{\partial E^{(t)}}{\partial a^{<t>}} + \frac{\partial E^{(t+1)}}{\partial a^{<t>}}).\,z^{<t>}\frac{\partial c^{<t>}}{\partial s^{<t>}}.\,(x^{<t>})^T$$

RNNVis

Major challenges

- RNNs maintain memory-like arrays called hidden states which store information extracted from a long input sequence
- The complex sequential rules embedded in texts are intrinsically difficult to be interpreted and analyzed.
- The semantic information in hidden states is highly distributed (how to interpret embedded information?).

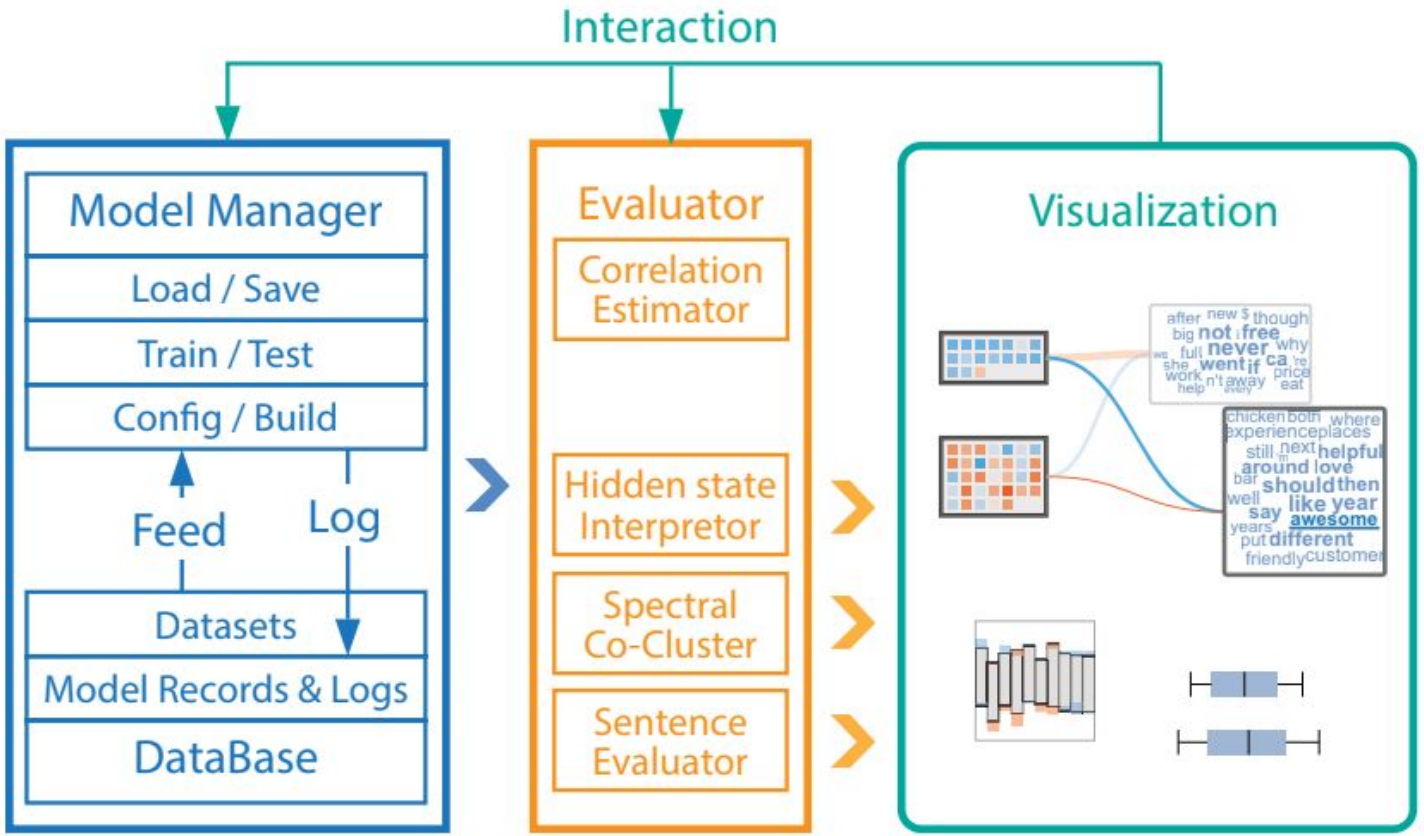
Visual Analytics

- Understand, compare, and diagnosis RNNs for general text-based NLP tasks.
- Explain a function of individual hidden state and textual information, based on their expected response.
- Sequence visualization to analyze the sentence-level behavior of RNNs.

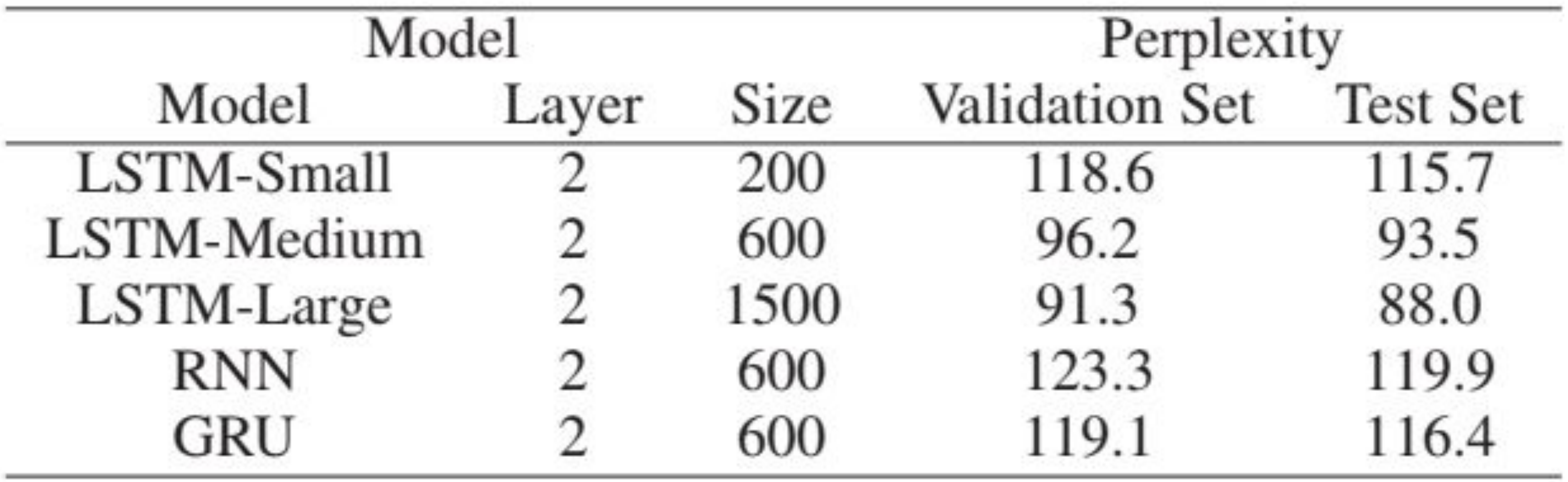
Requirement Analysis

- **R1 Clearly interpret the information captured by hidden states:** what kinds of words or grammars are captured and stored in a hidden unit?
- **R2 Provide the overall information distribution in hidden states:** how is the stored information differentiated and correlated across hidden states?
- **R3 Explore hidden states mechanisms at the sequence-level:** word embedding to a 2-D space? how does the internal memory updating mechanism behavior when dealing with sequences?
- **R4 Examine detailed statistics of individual states:** distribution of hidden state values or gate activations.
- **R5 Compare learning outcome of models:** what are the internal reasons that one model is better than the other?

RNNVis: System Architecture

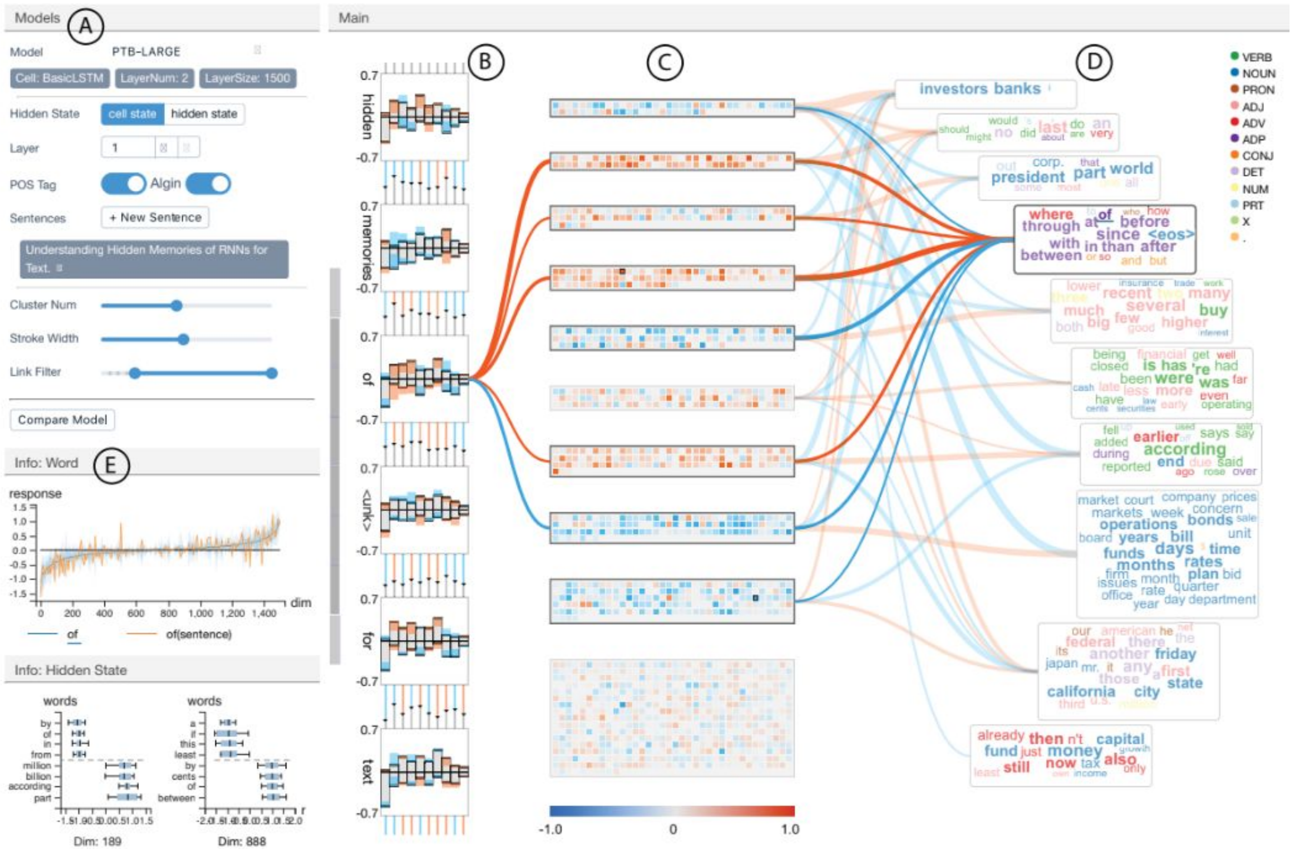


RNNVis: Models



Model			Perplexity	
Model	Layer	Size	Validation Set	Test Set
LSTM-Small	2	200	118.6	115.7
LSTM-Medium	2	600	96.2	93.5
LSTM-Large	2	1500	91.3	88.0
RNN	2	600	123.3	119.9
GRU	2	600	119.1	116.4

RNNVis:


Models
A
Main
B
C
D
E
Info: Word
Info: Hidden State
response
dim
of
of(sentence)
words
Dim: 189
Dim: 888
hidden
memories
of
<unk>
for
text
-1.0
0
1.0
VERB
NOUN
PRON
ADJ
ADV
ADP
CONJ
DET
NUM
PRT
X
.
investors banks
where through at of before since <eos> with in than after between
president part world
market court company prices operations bonds years bill funds days time months rates plan
california city state friday
already then n't capital fund money still now also

Model PTB-LARGE

Cell: BasicLSTM | LayerNum: 2 | LayerSize: 1500

Hidden State: [✓ cell state] hidden state

Layer: 1

POS Tag [✓] Algin [✓]

Sentences: + New Sentence

Understanding Hidden Memories of RNNs for Text.

Cluster Num

Stroke Width

Link Filter

Compare Model

RNNVis: Interpreting hidden states

- Tanh activation to get values (-1, 1).

$$\boldsymbol{h}^{(t)} = f(\boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{V}\boldsymbol{x}^{(t)})$$

- **U** is the output projection matrix

$$p_i = \text{softmax}(\boldsymbol{U}\boldsymbol{h}^{(T)})_i = \frac{\exp(\boldsymbol{u}_i^T\boldsymbol{h}^{(T)})}{\sum_{j=1}^{K}\exp(\boldsymbol{u}_j\boldsymbol{h}^{(T)})}$$

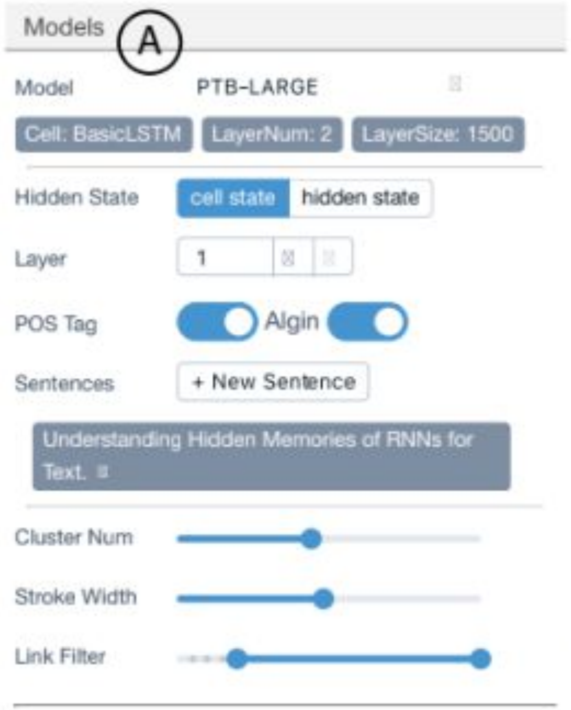
RNNVis: Control panel

- Select model trained in some datasets such as **Penn Tree Bank.**

- Select hidden layer to analyze

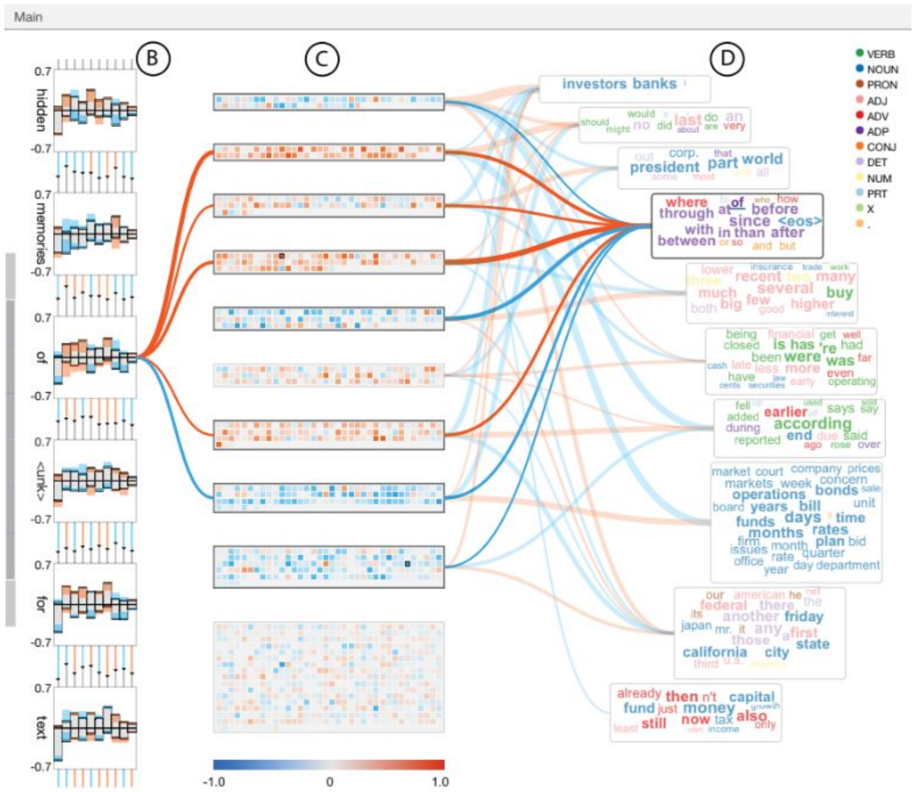
- Align and color words by POS

- Add new sentence to evaluate



RNNVis: Main view

- B) Show sequence visualization
- C) Show hidden states clusters
- D) Show word clusters
  Positions corresponds to cluster positions



RNNVis: Interpreting hidden states

- Contribution of word **t** to the predicted probability of class **i**.

$$\exp(\boldsymbol{u}_i^T \boldsymbol{h}^{(T)}) = \exp\left(\sum_{t=1}^{T} \boldsymbol{u}_i^T (\boldsymbol{h}^{(t)} - \boldsymbol{h}^{(t-1)})\right) = \prod_{t=1}^{T} \exp(\boldsymbol{u}_i^T \Delta\boldsymbol{h}^{(t)})$$

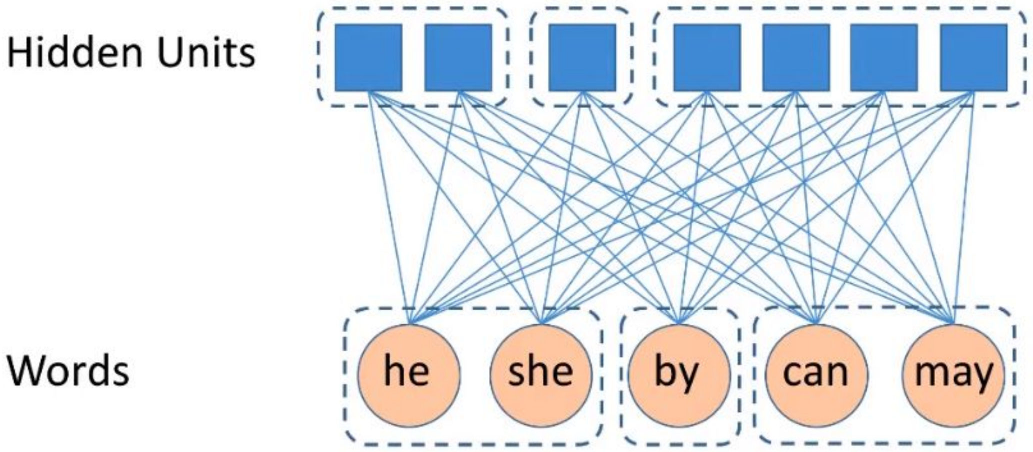
- Expected response to a word **w** computed by Adam’s Law.

$$s(\boldsymbol{x}) = E(\Delta\boldsymbol{h}^{(t)} \mid \boldsymbol{x}) = E(E(\Delta\boldsymbol{h}^{(t)} \mid \boldsymbol{x}, \boldsymbol{h}^{(t-1)}))$$

- Represents the relation between the **i-th** hidden state unit and **w**.

$$\hat{s}(\boldsymbol{x}) = \frac{1}{\sum_{\boldsymbol{x}^{(t)}=\boldsymbol{x}} 1} \sum_{\boldsymbol{x}^{(t)}=\boldsymbol{x}} \Delta\boldsymbol{h}^{(t)}$$

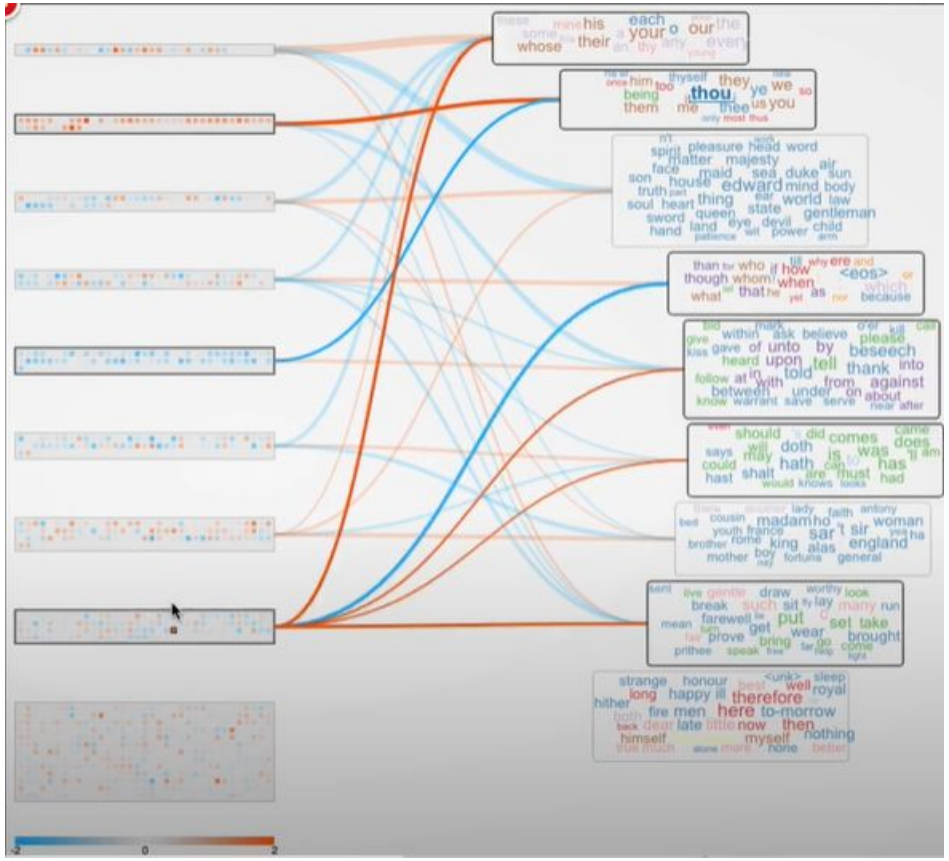
RNNVis: Spectral Co-Clustering



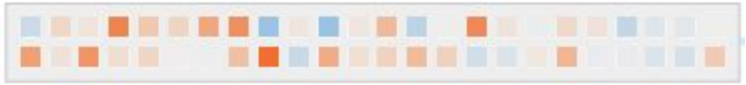
- Edges indicates the aggregate correlation between a word cluster **Wi** and a hidden state cluster **Hj**

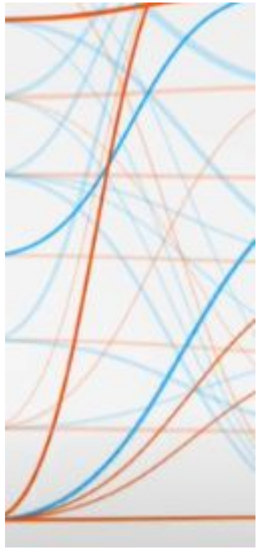
$$e(W_i, H_j) = \frac{1}{|W_i| \times |H_j|} \sum_{w_x \in W_i, h_y \in H_j} s(\boldsymbol{x}(w_x))_y$$

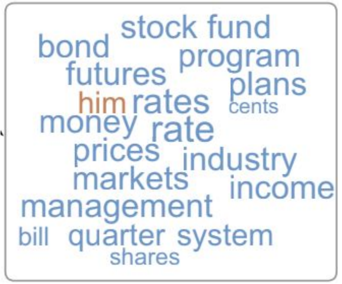
RNNVis: Main view



his their your our each whose
thou they we ye you them me thee us him thyself being
spirit pleasure head word matter majesty face maid sea duke air sun son house edward mind body truth soul heart thing ear world law state queen gentleman sword land eye devil child hand power
than who how when though whom what that <eos> which because
within ask believe please give gave of unto by beseech heard upon told tell thank into follow at with in from against between under on about know warrant save serve near after
should did comes does says will may doth is was hath has could hast shalt are must had
lady faith antony cousin madam woman youth france sir brother king alas england mother boy general fortune
sent draw look break such sit lay many run farewell put set take mean get prove wear bring go brought prithee speak come
strange honour sleep long happy ill therefore royal hither fire men here to-morrow dear late now then himself myself nothing none better
-2
0
2







RNNVis: Glyph Design of Sequence (sentence or paragraph) Nodes

- **Aggregate Information:** The sums of positive and negative hidden units in cluster **H_i**:

$$\alpha_{i+}^{(t)} = \sum_{h_j \in H_i, h_j > 0} h_j^{(t)}, \qquad \alpha_{i-}^{(t)} = \sum_{h_j \in H_i, h_j < 0} h_j^{(t)}. \qquad \alpha_i^{(t)} = (\alpha_{i+}^{(t)}, \alpha_{i-}^{(t)})$$

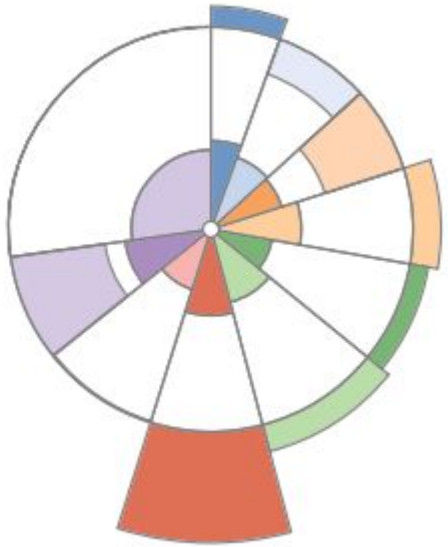
- **Updated information:** High value of the sum means that hidden state cluster **H_i** is highly correlated to **x**.

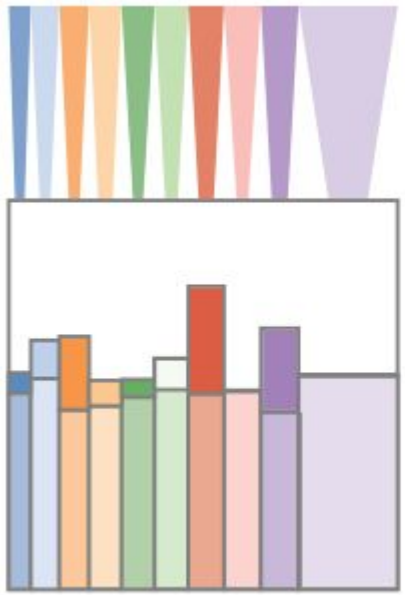
$$|\Delta\alpha_{i+}^{(t)} + \Delta\alpha_{i-}^{(t)}|$$

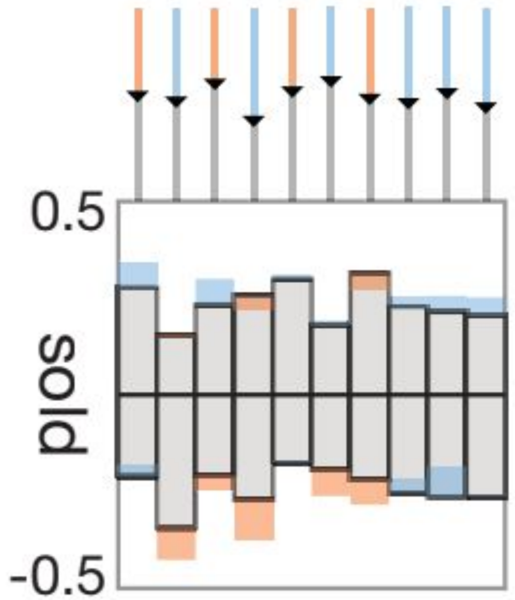
- **Preserved information:** Measures how much information in a hidden state cluster **H_i** has been retained after processing a new input **x_t**

$$\beta_i^{(t)} = \sum_{h_j \in H_i} |h_j^{(t-1)}| \min(1, \max(0, \frac{h_j^{(t)}}{h_j^{(t-1)}}))$$

RNNVis: Glyph Design of Sequence (sentence or paragraph) Nodes





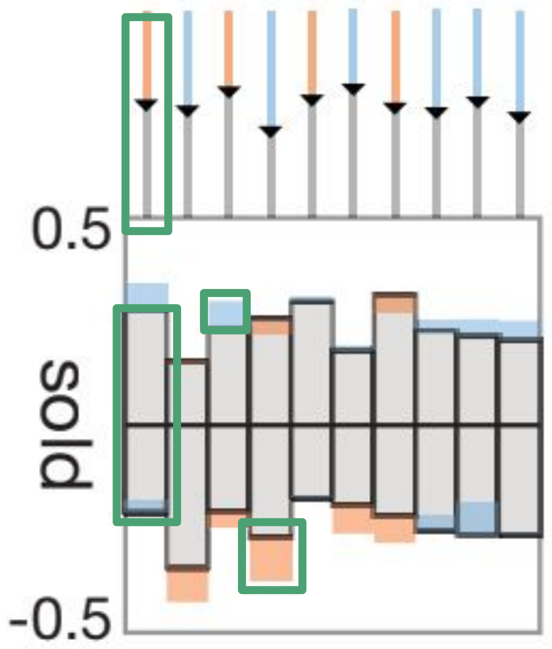


RNNVis: Glyph Design of Sequence (sentence or paragraph) Nodes

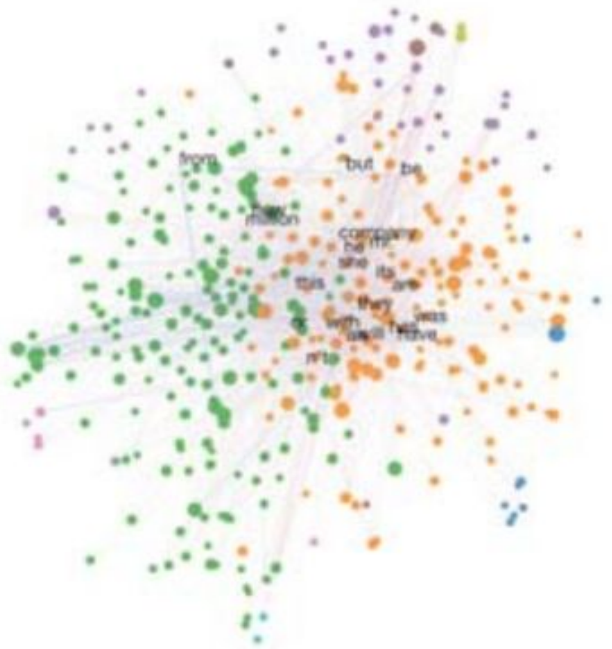
Aggregate Information $\alpha_i^{(t)}/|H_i|$

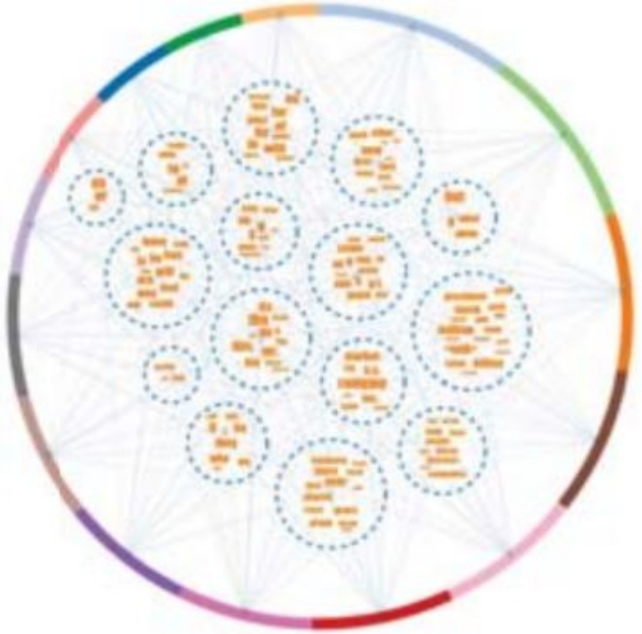
Updated Information $\Delta\alpha_{i-}^{(t)}/|H_i|$ $\Delta\alpha_{i+}^{(t)}/|H_i|$

Preserved information $\beta_i^{(t)}/(\alpha_{i+}^{(t)} - \alpha_{i-}^{(t)})$



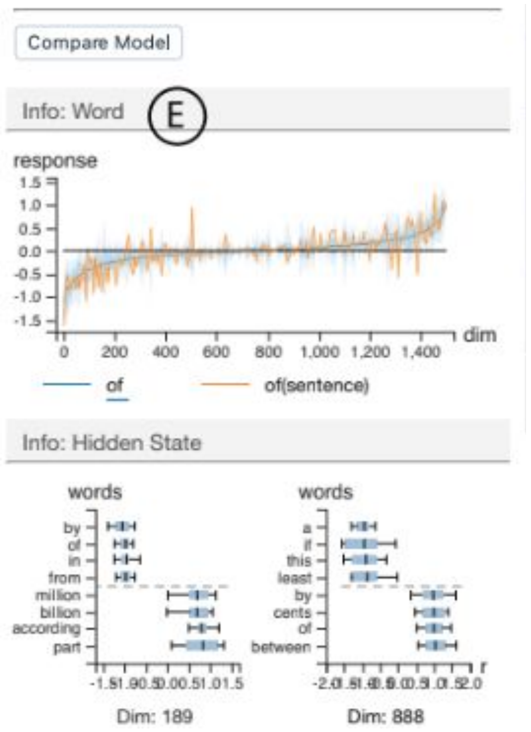
RNNVis: Glyph Design of Sequence (sentence or paragraph) Nodes



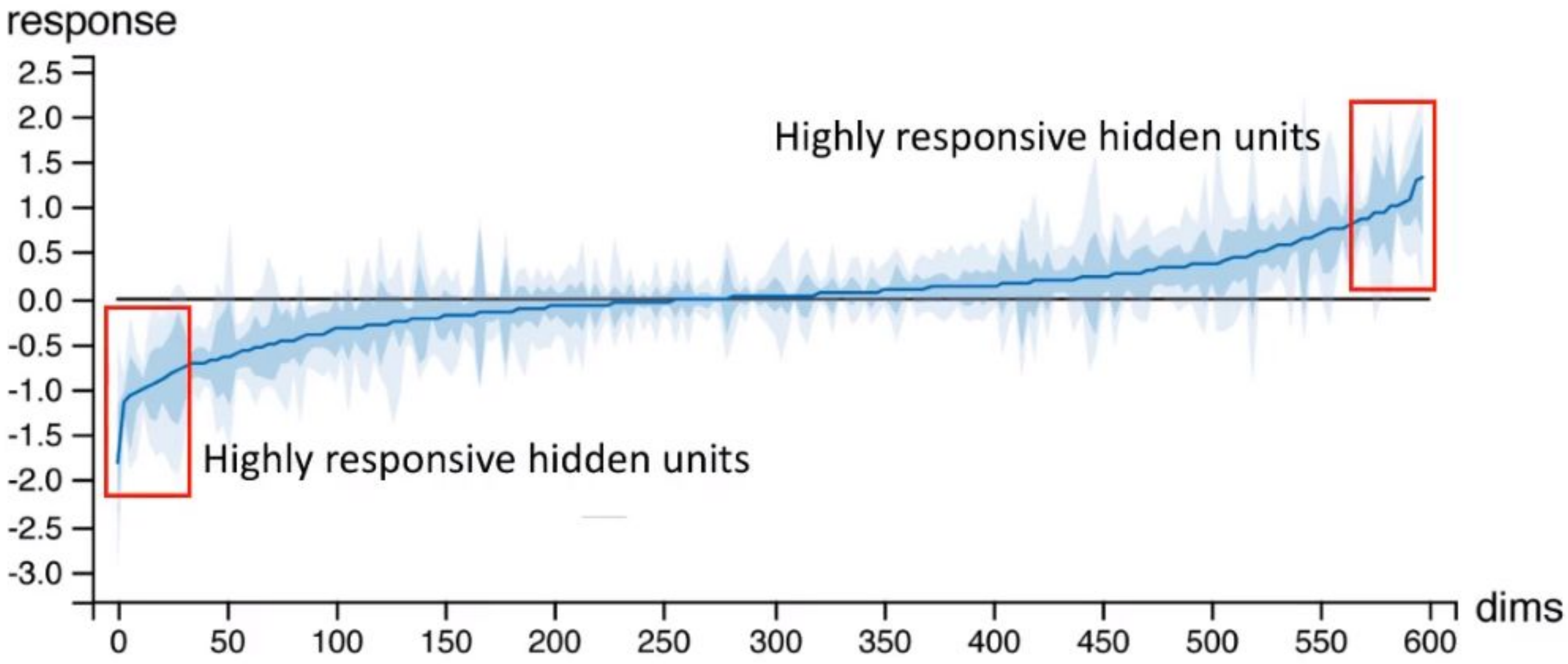


RNNVis: Detail view

- Explore the distribution of model's responses to particular words

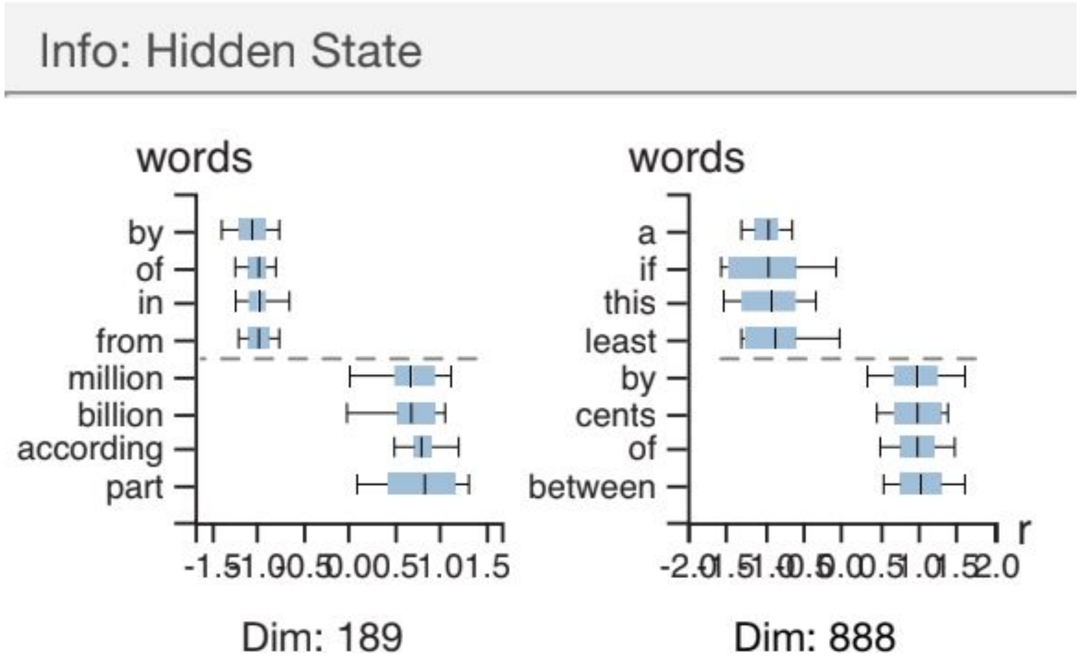


RNNVis: Detail view



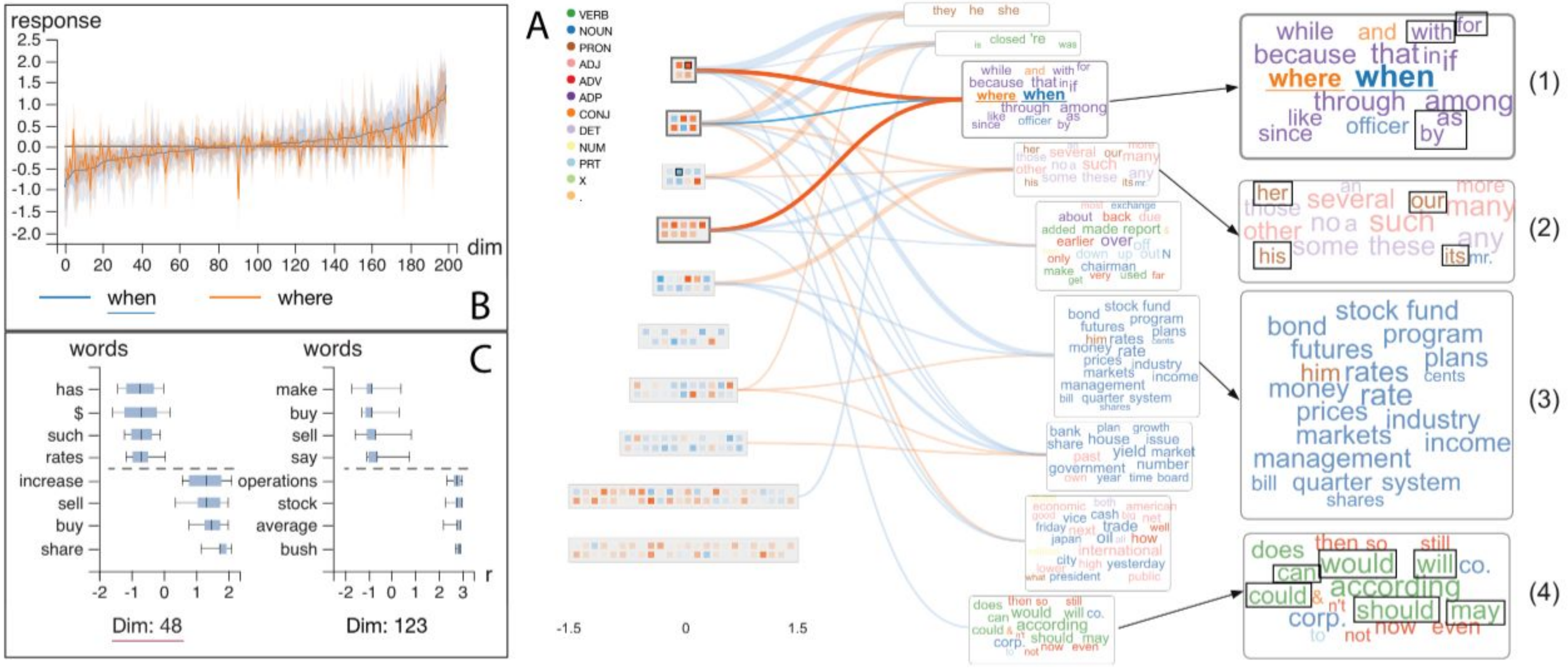
Response statistics of the hidden state vector to the word “he”

RNNVis: Detail view

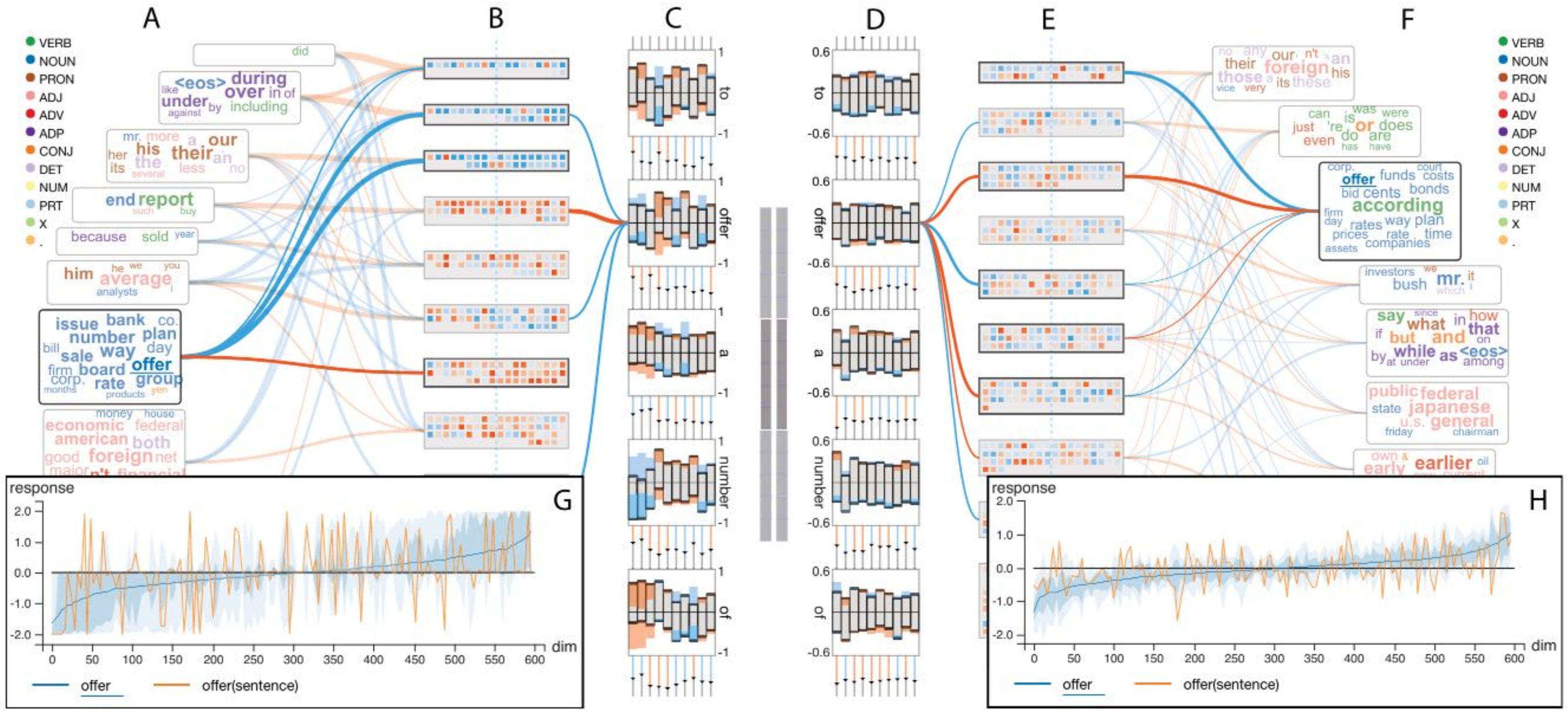


Case Use

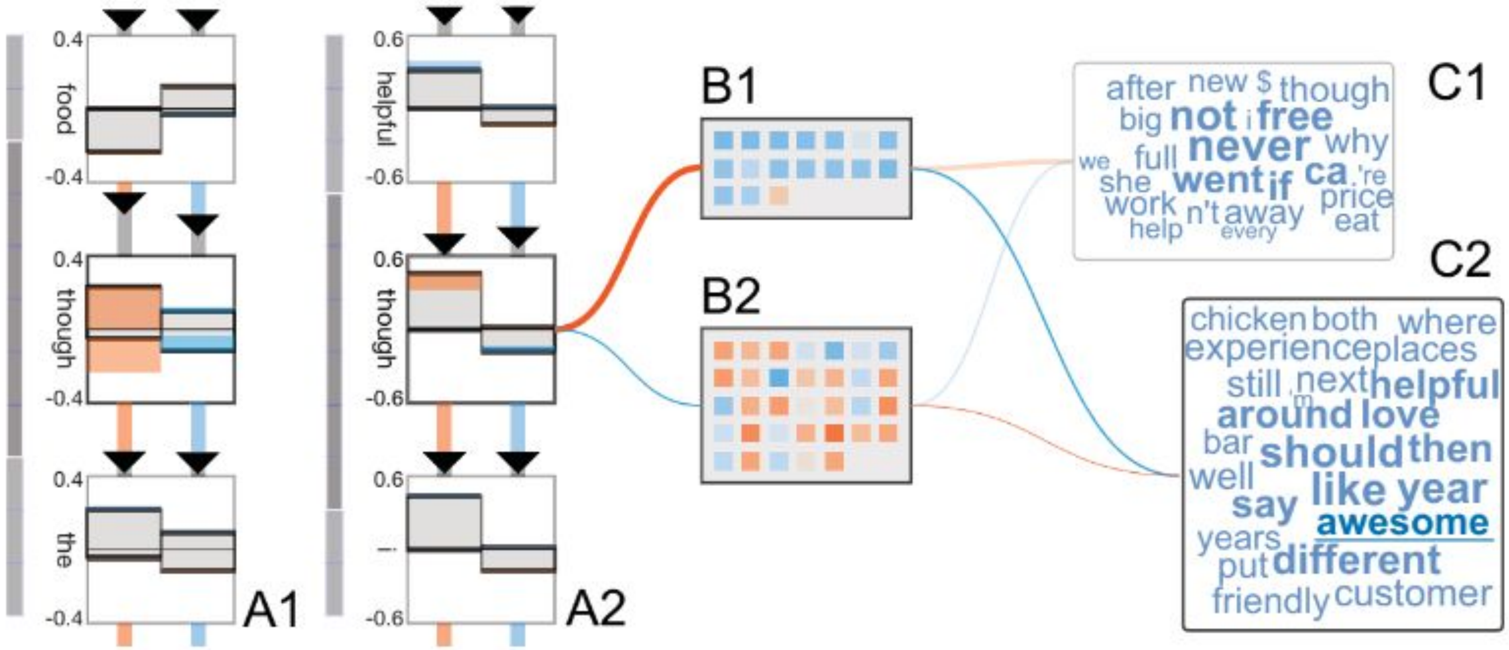
Language Modeling: Penn Tree Bank (PTB)



Language Modeling: Penn Tree Bank (PTB)



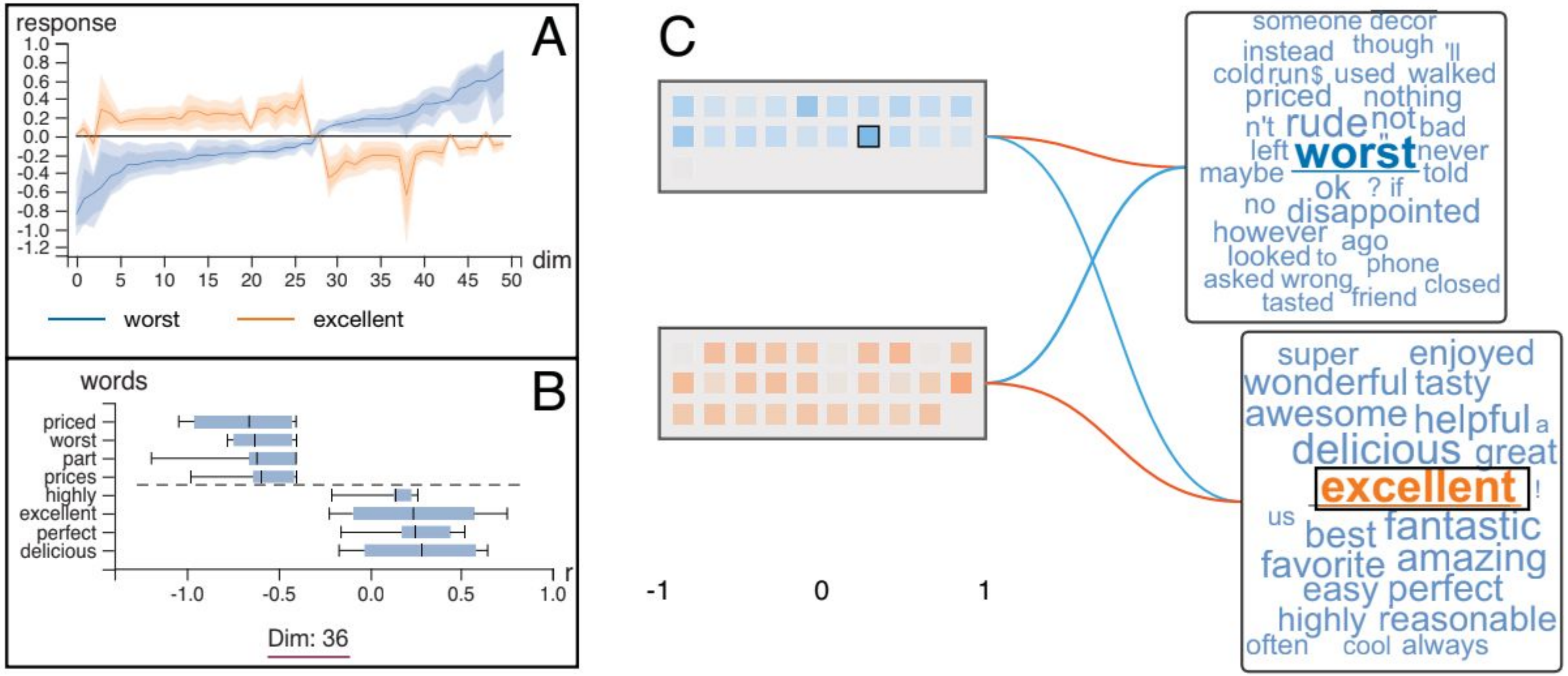
Sentiment Analysis: Yelp Data Challenge



0.4
food
-0.4
0.4
though
-0.4
0.4
the
-0.4
A1
0.6
helpful
-0.6
0.6
though
-0.6
0.6
i
-0.6
A2
B1
B2
C1
after new $ though big not i free never why we full she went if ca 're work n't away price help every eat
C2
chicken both where experience places still next m helpful around love bar should then well say like year awesome years put different friendly customer

"I love the food though staff is not helpful" vs "The staff is not helpful though i love the food"

Sentiment Analysis: Yelp Data Challenge



"I love the food though staff is not helpful" vs "The staff is not helpful though i love the food"

References

- LSTMVis
- RNN review
- Understanding hidden layers