How LSTM Encodes Syntax: Exploring Context Vectors and Semi-Quantization on Natural Text

Chihiro Shibata (Tokyo University of Technology)

Kei Uchiumi (Denso IT Laboratory)

Daichi Mochihashi (The Institute of Statistical Mathematics)

Introduction

- LSTMs have been used in wide range of NLP tasks:
  - Machine translation, text generation, etc.
- LSTM Language Model (LSTM-LM) is the most fundamental architecture for those applications.
  - It is not yet completely clear how syntactic information is represented in it.

- What is the purpose of this research?
  - Understanding internal representations of LSTM-LMs w.r.t. syntactic information.
    - Empirical approach: Real data (plain text) + syntactical annotation.
    - Details of representations in each internal vector inside LSTM are investigated.
  - NOT about BERT's representation.
    - nor comparison to BERT

Outline

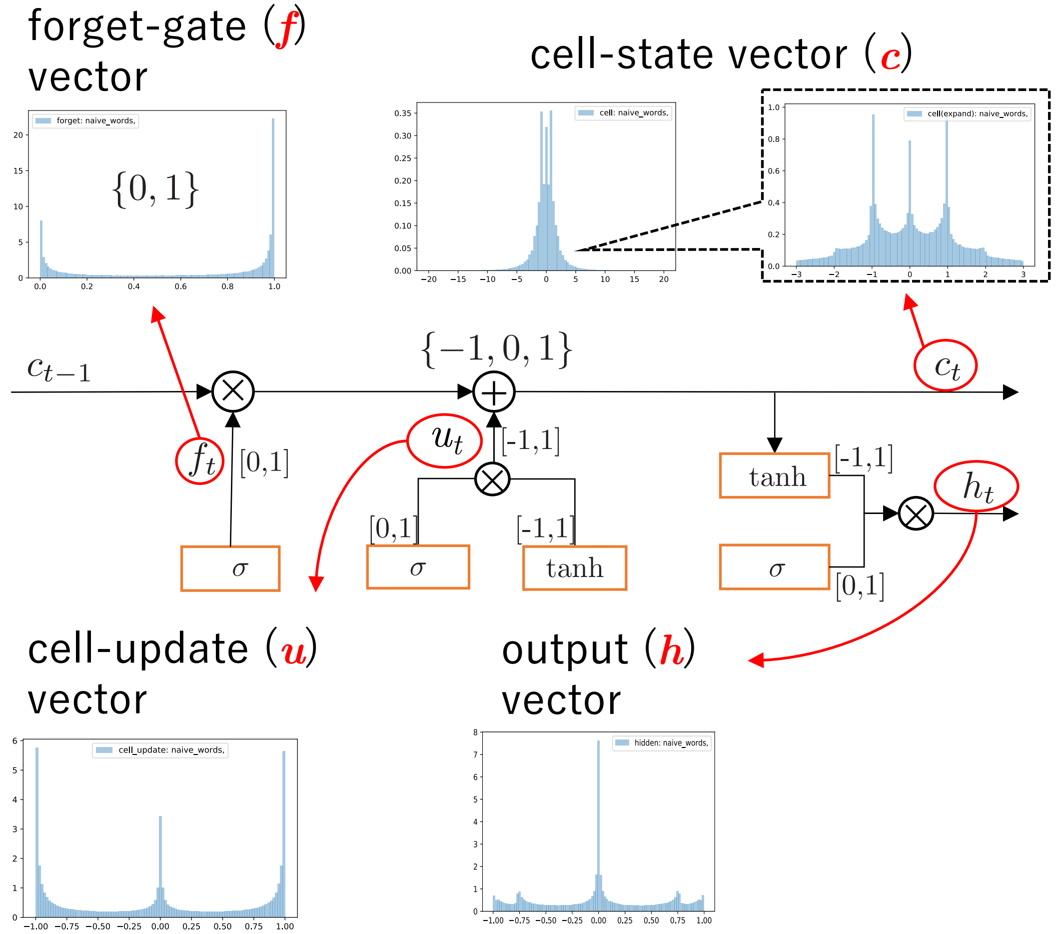
1. We investigate the distributions of the elements of the internal vectors.
    - empirically show that their distributions are approximately quantized (**<u>Semi-quantization</u>**).

2. **<u>Cell-state vectors</u>** ($\boldsymbol{c}$) are investigated using several datasets, some of which are Dyck-languages.
    - How the semi-quantization relates to the representation in $\boldsymbol{c}$.

3. **<u>Cell-update vectors</u>** ($\boldsymbol{u}$) are focused and
    - showed to have important role in representing syntactic information.

Semi-Quantization of Internal Vectors and Statistics of each Element of them

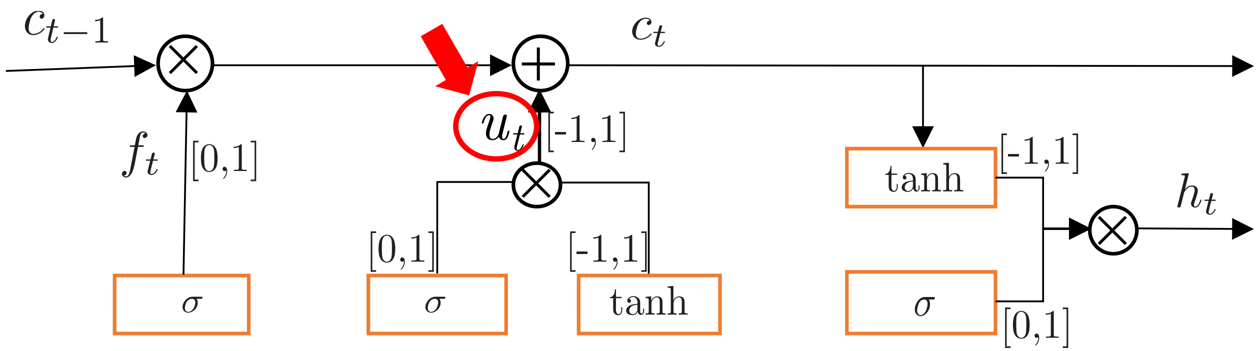
- Right figure: learning results of a single-layered LSTM-LM using plain texts (WSJ) are shown.
- Simple but important facts:
  - Each element is approximately quantized (because LSTM is designed so).
  - Internal vectors such as cell-state and output have characteristic distributions that have different peaks.
- Datasets and learning methods such as dropout basically are independent to the above characteristics.

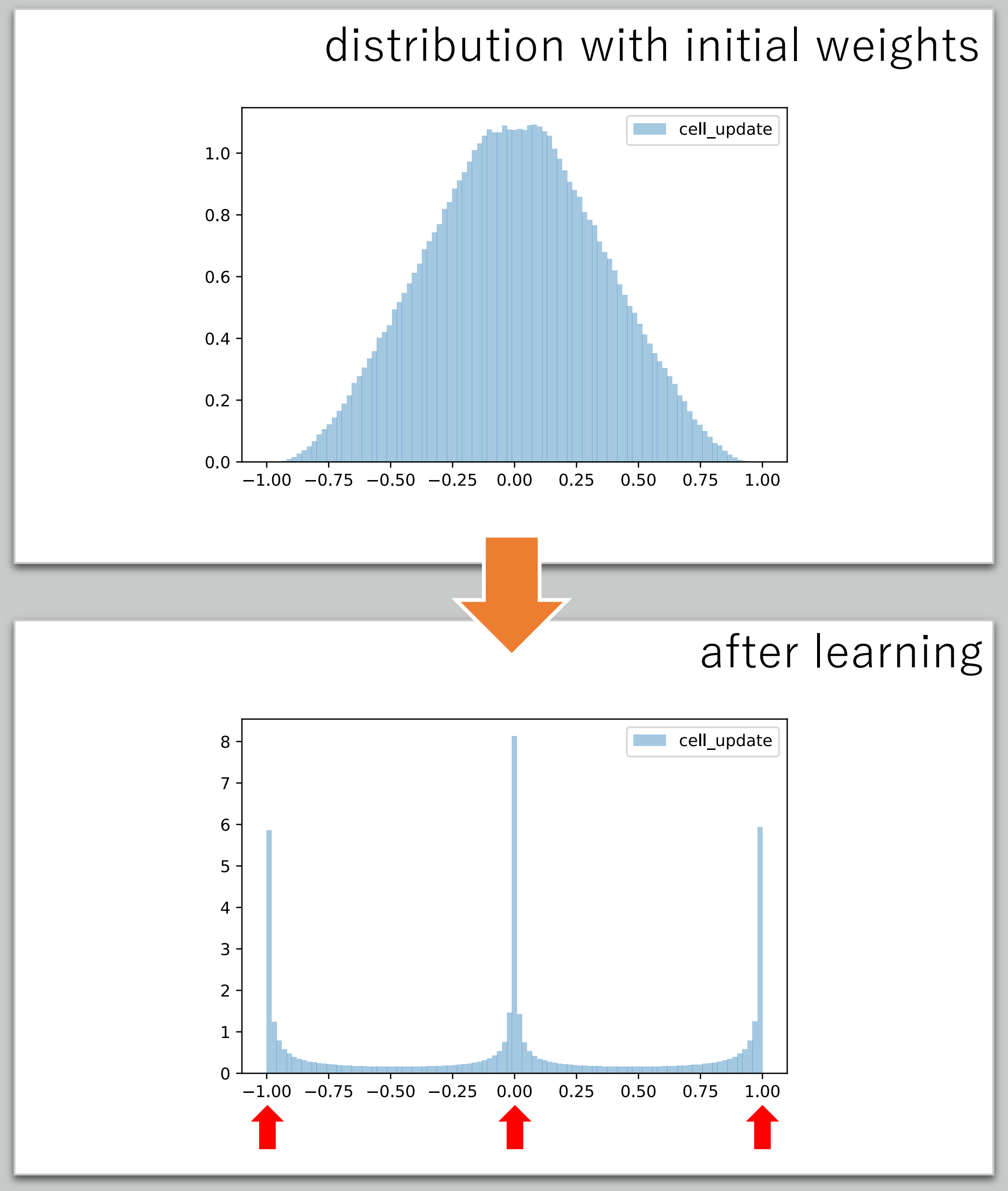


We look into each distribution that the elements of each internal vector has.

Distribution of elements of cell-update vectors (u)

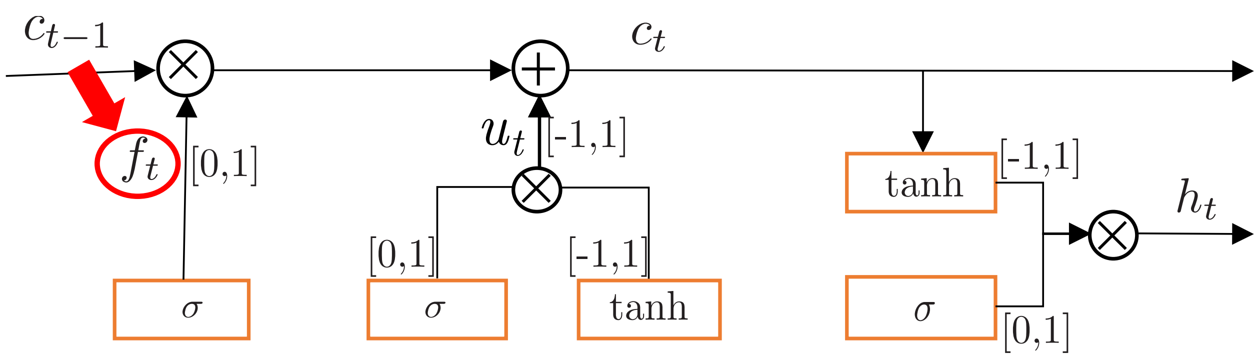
- Distributions dramatically changes through learning.
- u is semi-quantized into {-1,0,1}.
- has important rules for syntactic representation as shown later.

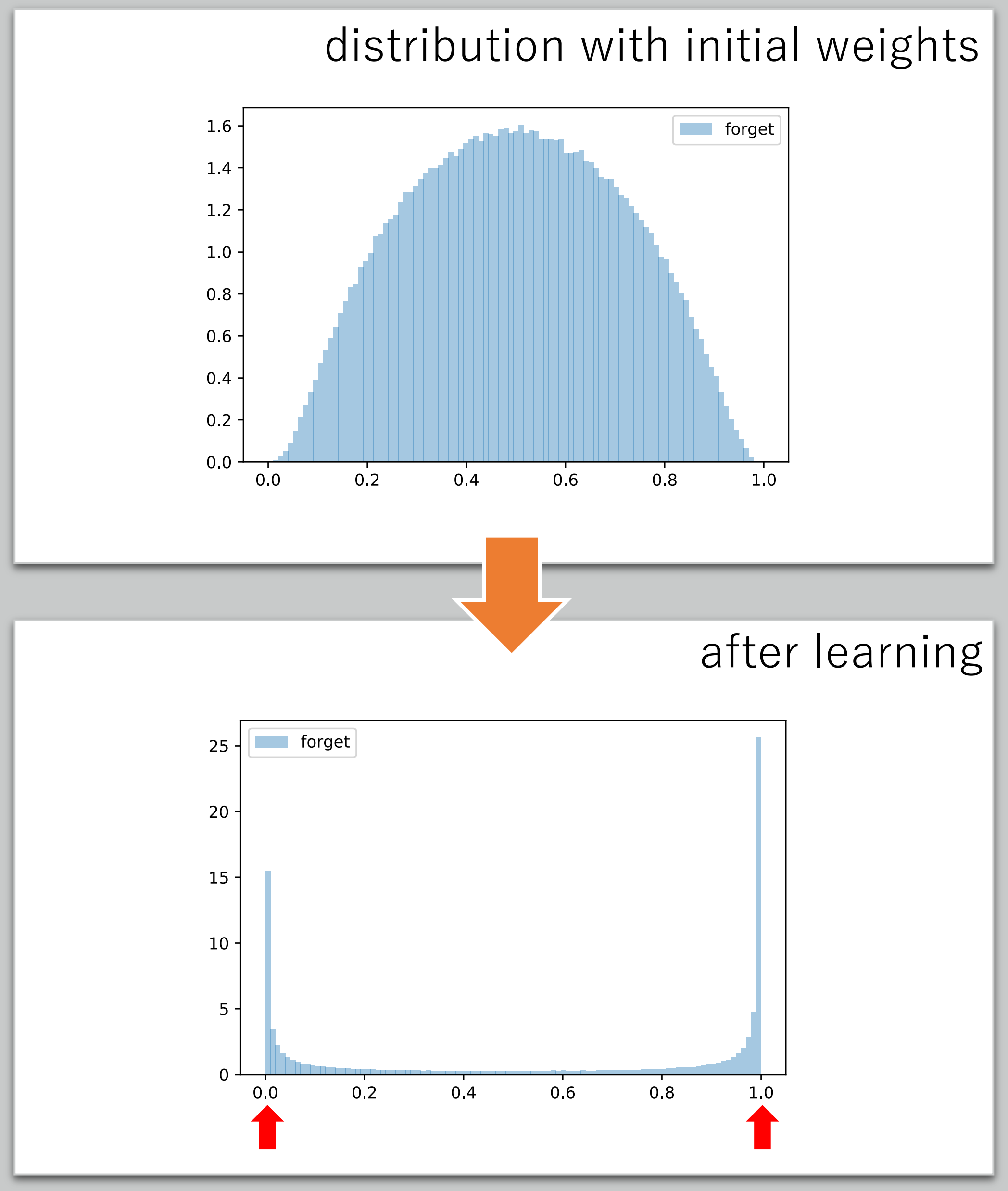




Distribution of elements of forget-gate vectors (f)

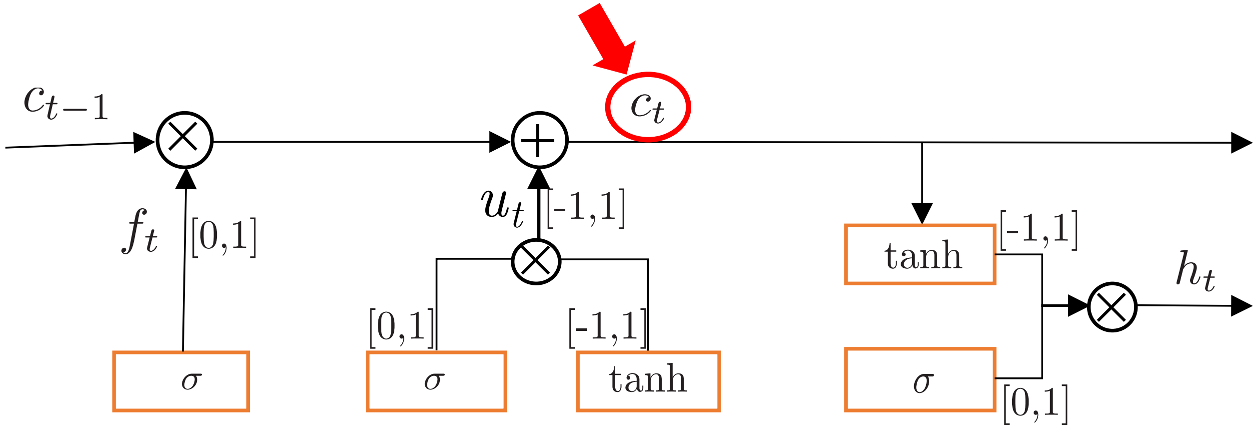
- Distributions of elements of $\boldsymbol{f}$ are binarized into {0,1} values.

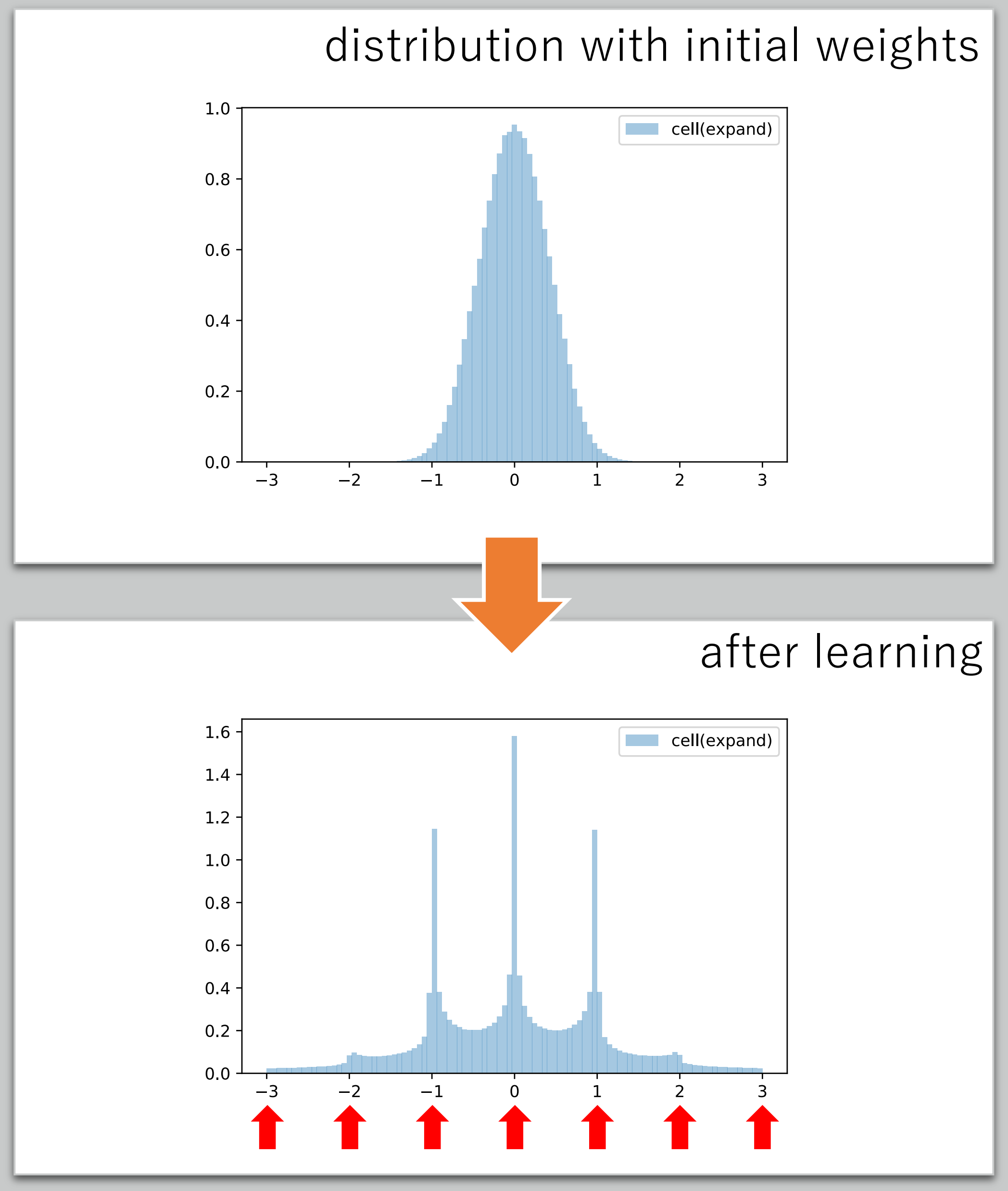




Distribution of elements of cell-state vectors (c)

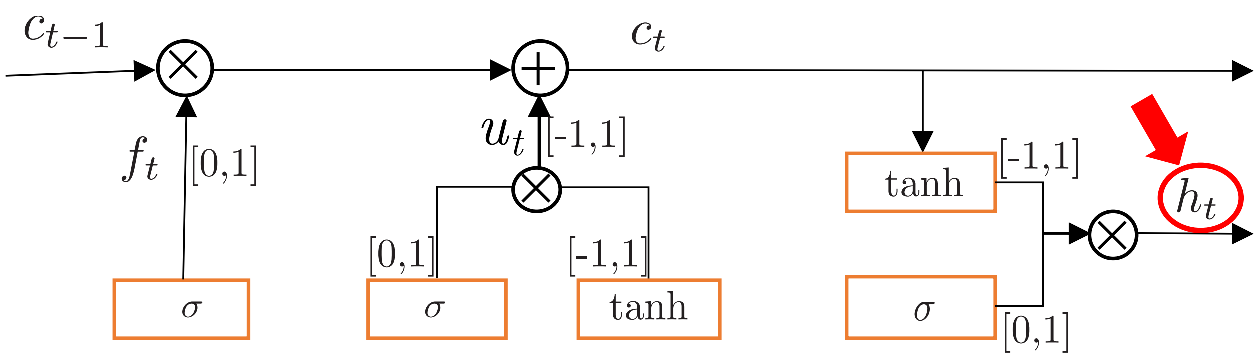
- We can observe peaks in the integer values:
- result of accumulating $\boldsymbol{u}$ vectors.

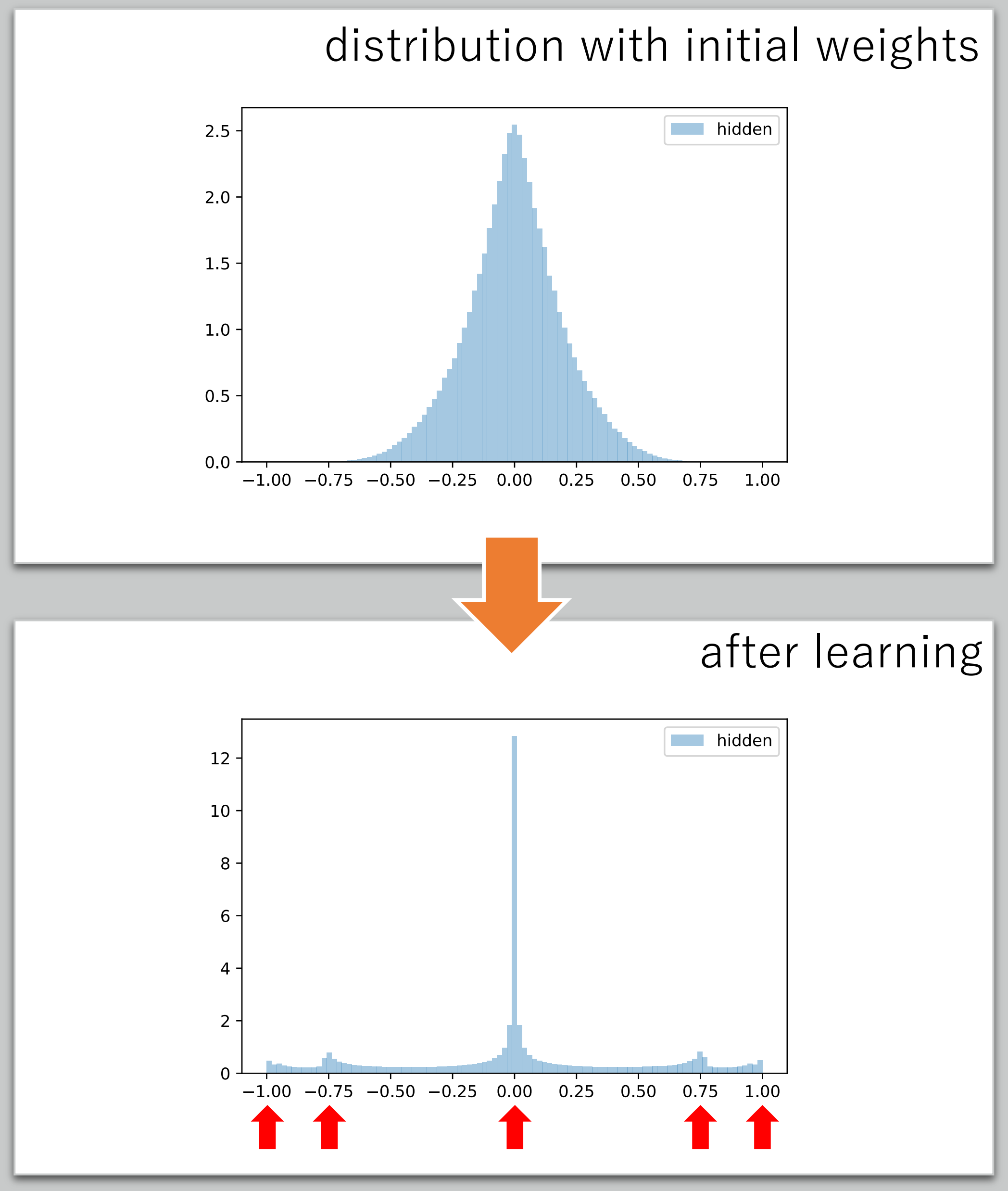




Distribution of elements of output (=hidden) vectors (*h*)

- Large peak around 0.
- Small peaks at {-1, -0.75, +0.75, +1}.





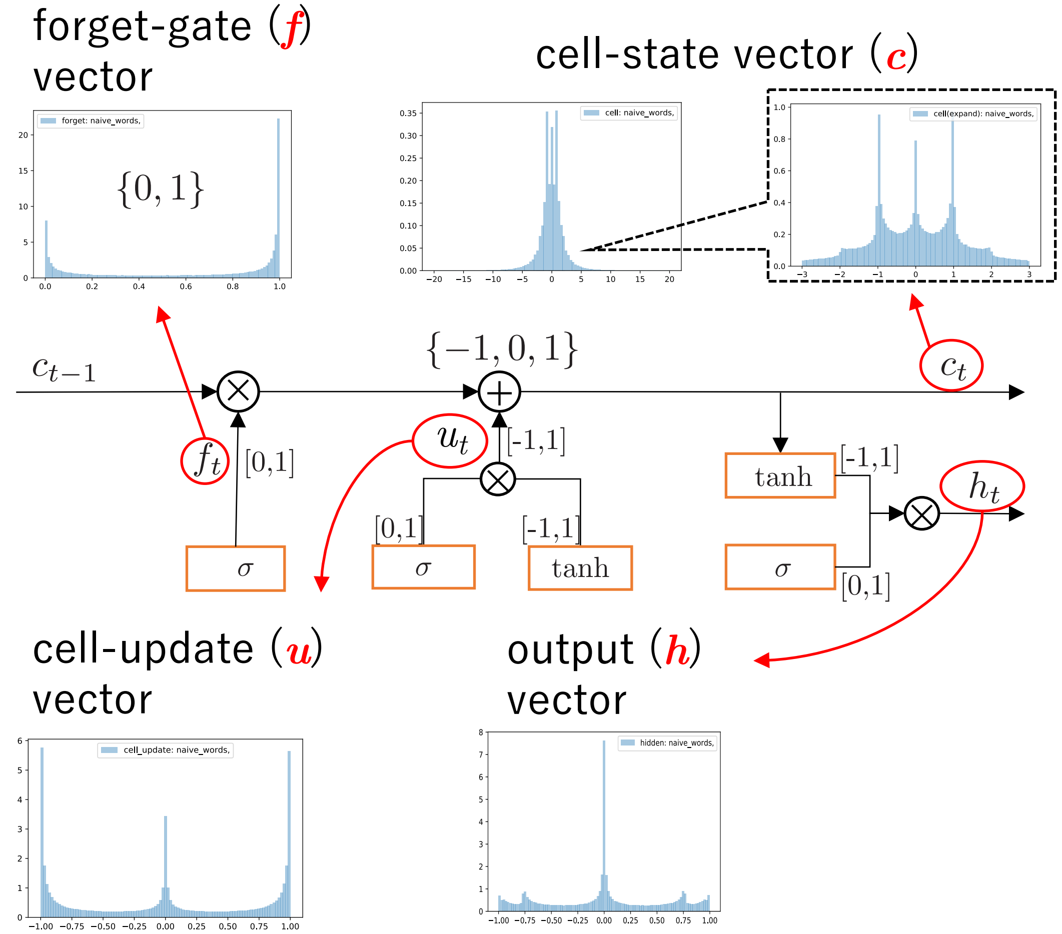
Semi-quantization of internal vectors and statistics of each element of vectors

- Question:

  This kind of semi-quantization

  - is just a result of activation functions, and thus there is no contribution to encode syntax?
  - or has a certain role for learning syntax?

- Through experiments:
  - We investigate the representation in $\boldsymbol{c}$ and its relation to <u>the nesting depths of the phrase structures</u>.
  - Models are learned from several types of data.



Experiments: Target-dataset to learn

- Making Dyck-like data by adding parentheses to texts in PTB-WSJ.
- Four types of data:

1. **Paren** : '(' and ')' without words,

   `( ( ) ( ) ( ) )`

2. **Paren+W** : '(' and ')' + words,

   `( ( a ) ( nonexecutive ) ( director ) )`

3. **Tag** : '(T' and 'T)' without words, where T is a nonterminal symbol.

   `(NP (DT NP) (JJ JJ) (NN NN) NP)`

4. **Tag+W** : '(T' and 'T)' + words:

   `(NP (DT a NP) (JJ nonexecutive JJ) (NN director NN) NP)`

5. **Words** : plain text.

   `a nonexecutive director`

Learning results and accuracies

Check accuracies for ① the balancing of parentheses and
② kinds of tags (implying orders of tags) :

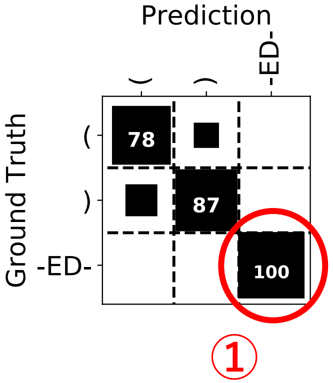
① LSTM-LM predicts EOS with 100% (almost no mistake).

② It predicts kinds of phrases of ")" with >95% (slight mistake).

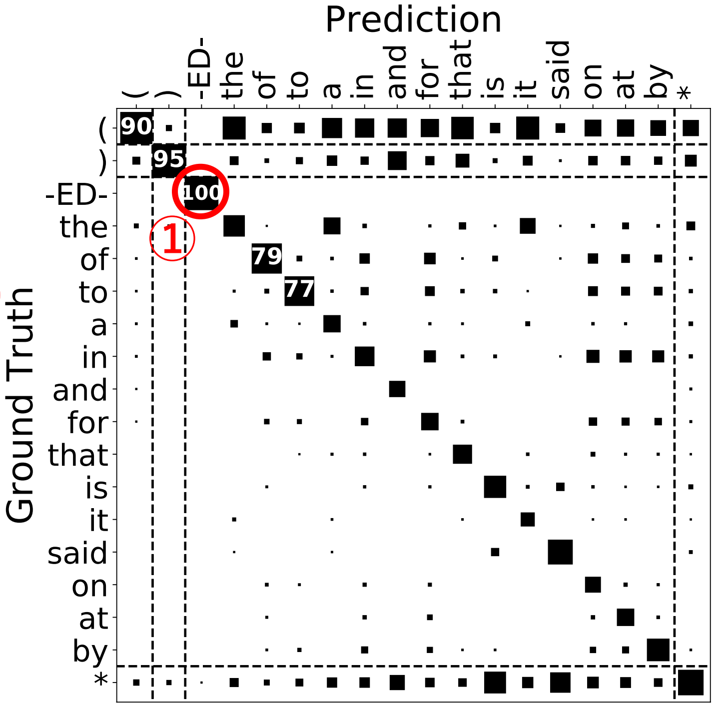
End Of Sentence

Dataset	BOP	EOP	EOS	Words
Paren	0.77	0.87	**1.00**	–
Paren+W	0.90	0.96	**1.00**	0.78
Tag	0.87	0.93	**1.00**	–
Tag+W	0.89	0.96	**1.00**	0.86
Words	–	–	–	0.49

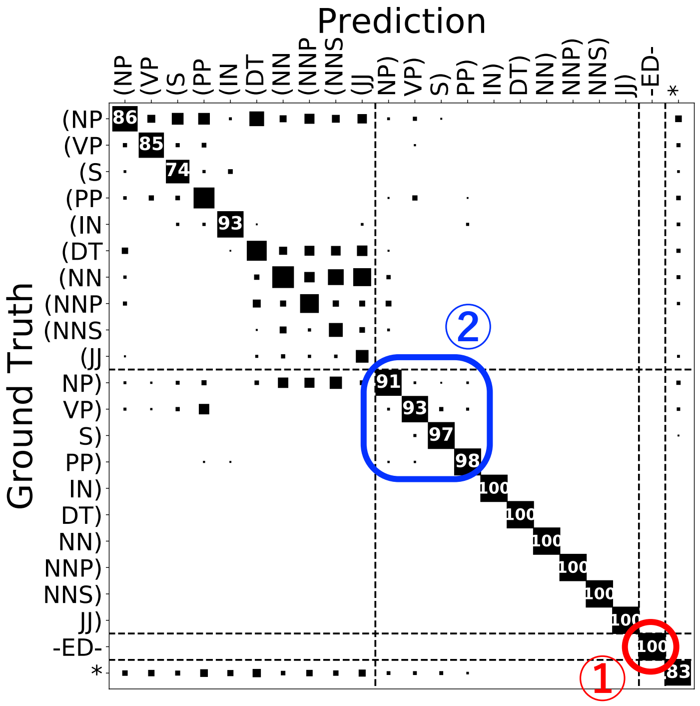
Paren



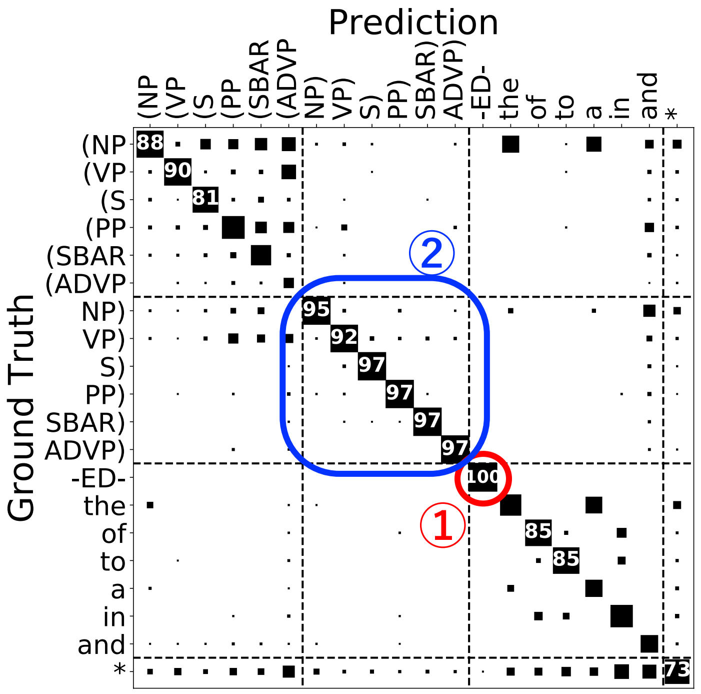
Paren+W



Tag

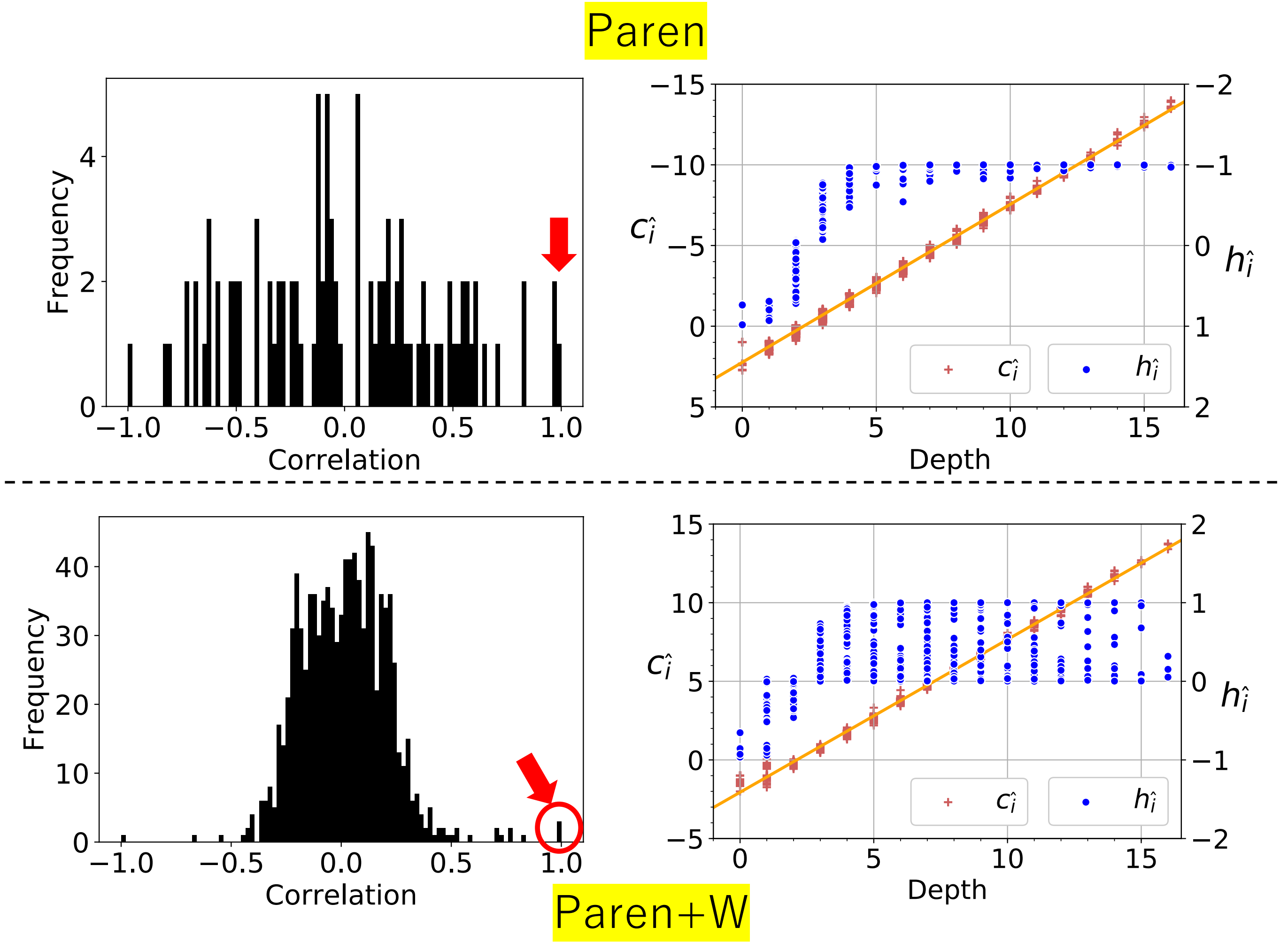


Tag+W



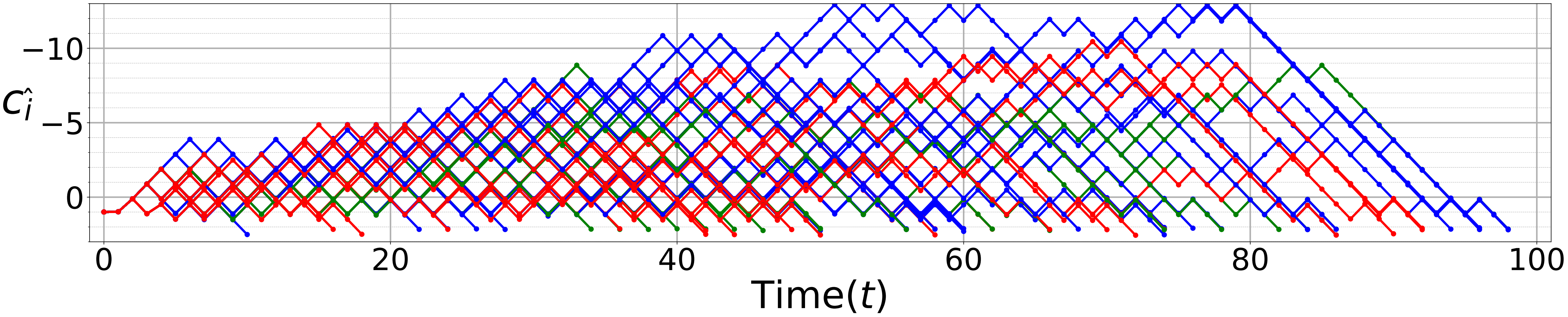
Embedding of nesting depth on Paren and Paren+W data

- In cell-state vector (c):
  - Elements whose correlation coefficient is 1.0 with respect to the nesting depth.
  - Both for Paren and Paren+W.
- LSTM counts the nesting depth of the parentheses through such elements.



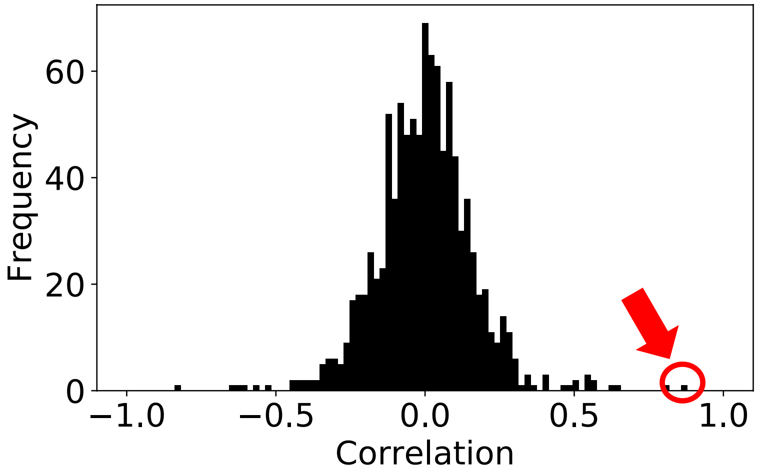
Visualizing of count of nesting depths by a single element of $\boldsymbol{c}$

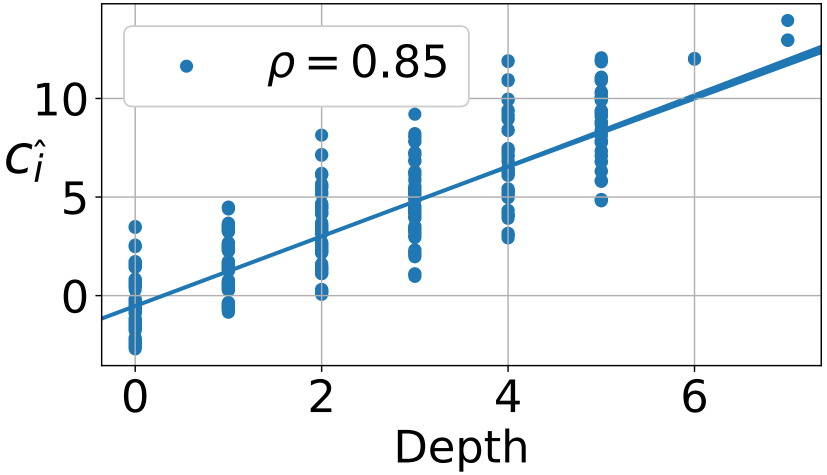
- For Paren data, we can observe a clear lattice for some single element.
- As the height of each step of the lattice is 1, we can know that $\boldsymbol{c}$, $\boldsymbol{u}$, $\boldsymbol{f}$ are completely quantized to natural numbers.

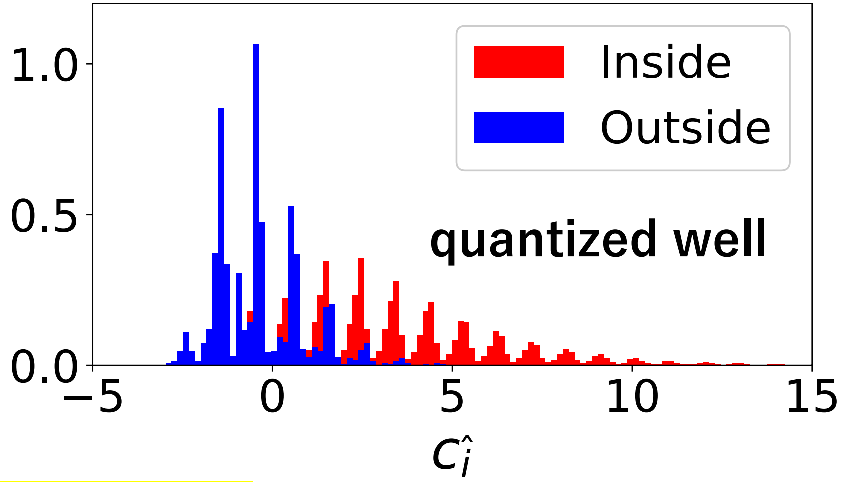


Embedding of nesting depth of each tag on Tag and Tag+W data

- There is a single element that has high correlation.
  - the highest correlated elements : 0.96 for NP, and 0.85 for VP.
- However, correlation is not perfect.
  - Element of $\boldsymbol{c}$ is quantized well but doesn't corresponds to the nesting depth of VP perfectly.



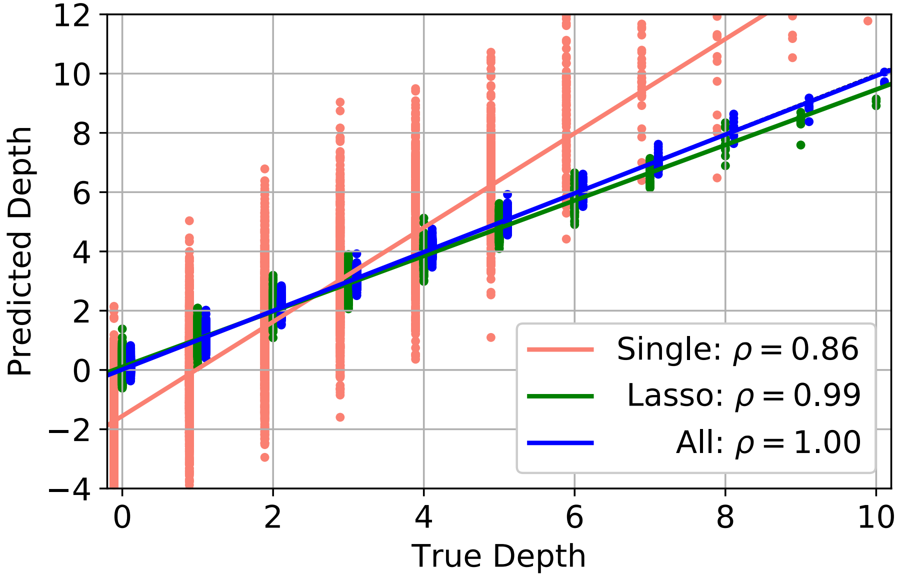




cell-state and the nesting depth of VP (Tag+W)

Representation in subspace (linear sum of elements)

- we can find a good linear sum so that the correlation coefficient can be almost 1.

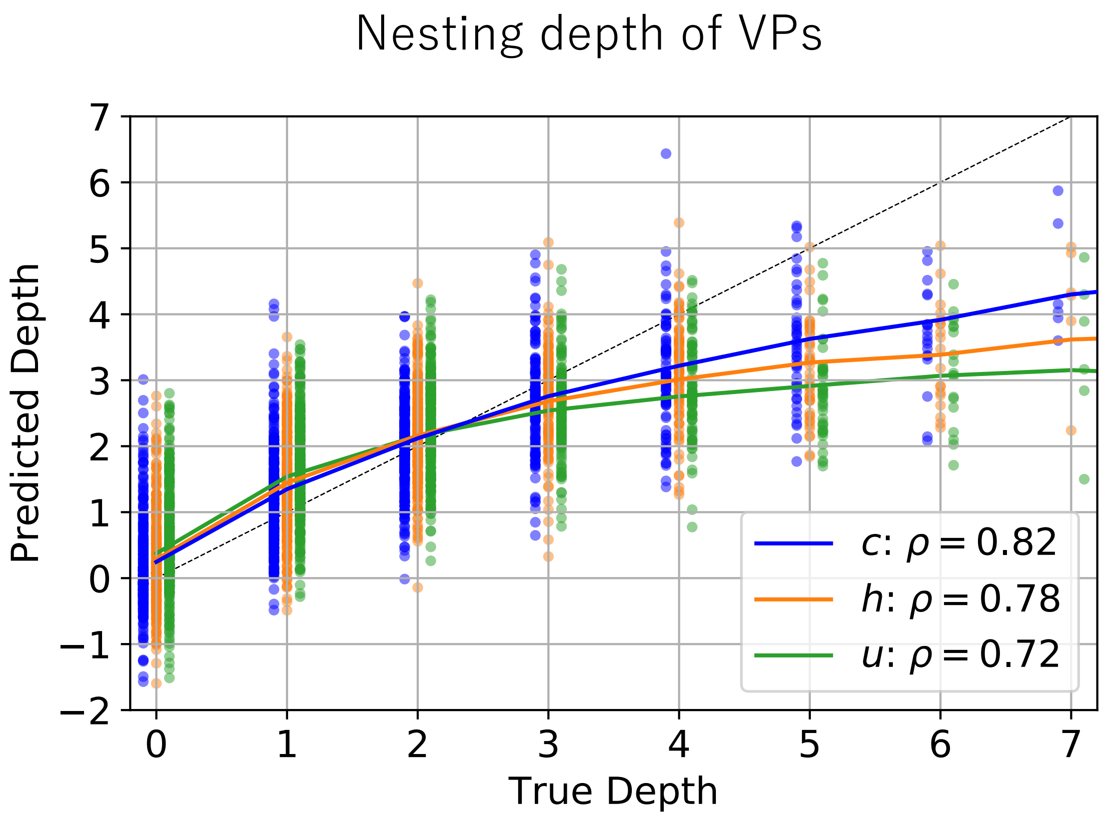


Dataset Tag			Dataset Tag+W			C
Acc	#nnz	ratio	Acc	#nnz	ratio	
0.996	82	41%	**0.9996**	134	13%	3×10^{-3}
0.994	56	28%	0.9992	100	10%	1×10^{-3}
0.991	34	17%	0.998	71	7%	3×10^{-4}
0.98	21	11%	0.991	51	5%	1×10^{-4}
0.96	8	4%	0.97	27	2.7%	3×10^{-5}
0.91	**5**	2.5%	0.87	**12**	1.2%	1×10^{-5}

Embedding of nesting depth using plain text

- Correlation coefficient is high:
  0.82 for VP using linear sum of $\boldsymbol{c}$

- It is not possible to obtain a complete correlation such that all plots are almost on a straight line.



Summary so far and Further Question

Summary:

- For Paren data, in $\boldsymbol{c}$, there is a completely quantized element that acts as a counter of the nesting depth of the parentheses.
- For Tag data, a linear sum of the elements of $\boldsymbol{c}$ can act as a counter of the nesting depth.
- For plain text, we cannot find such a clear counter, but find a highly correlated direction in $\boldsymbol{c}$.

Question:

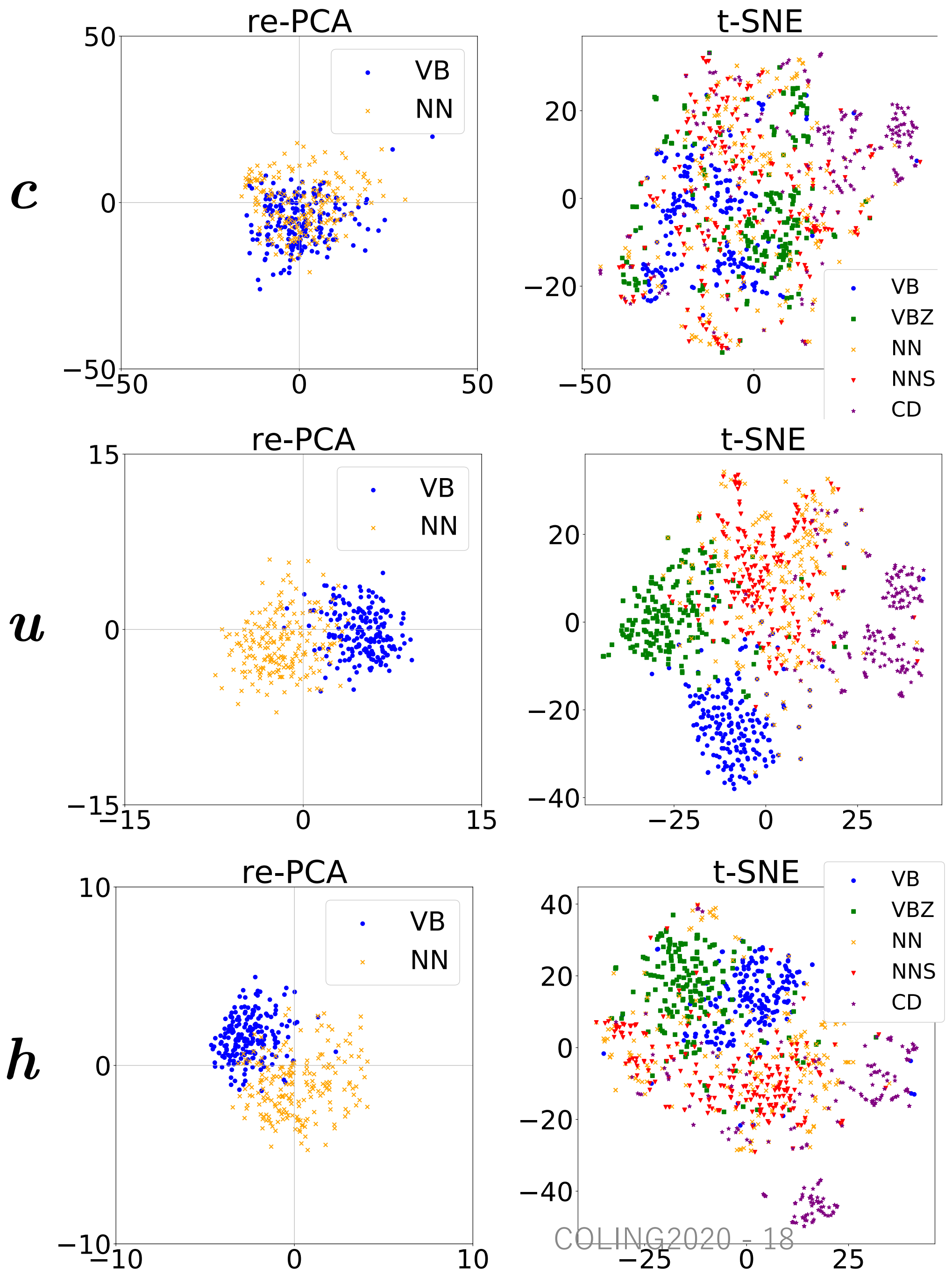
- For plain text, can we find any clusters that represents triggering the nesting of the phrase structure, which should be POS such as nouns and verbs?

Representation of Parts of Speech

- comparing $\boldsymbol{c}$, $\boldsymbol{u}$, $\boldsymbol{h}$:
  - $\boldsymbol{c}$ has accumulated context information.
  - $\boldsymbol{u}$ has delta that triggers contexts.
  - $\boldsymbol{h}$ has information to predict a next word.

  $\boldsymbol{u}$ should represent POS most clearly.
- visualizing POS clusters:
  - Vectors for the same word are averaged.
  - Clusters of {VB, VBZ, NN, NNS,CD} are obtained most clearly in $\boldsymbol{u}$.

NN, NNZ, VB, VBZ : noun singular, plural, verb sin., plu.
CD : numbers



list of similar words : understanding roles of internal vectors

- comparison of vector similarities between c and u.
- For $\boldsymbol{u}$, syntactically similar words tend to be listed.
- For $\boldsymbol{c}$, co-occurrence words tend to be listed.

"her"				"his"				"an"				"a"			
$\boldsymbol{c}$	sim.	$\boldsymbol{u}$	sim.	$\boldsymbol{c}$	sim.	$\boldsymbol{u}$	sim.	$\boldsymbol{c}$	sim.	$\boldsymbol{u}$	sim.	$\boldsymbol{c}$	sim.	$\boldsymbol{u}$	sim.
his	0.70	his	0.39	the	0.74	the	0.43	a	0.71	a	0.31	the	0.76	the	0.43
mother	0.68	my	0.33	's	0.73	their	0.39	the	0.68	the	0.27	modest	0.76	another	0.36
playing	0.67	the	0.28	a	0.72	her	0.39	initial	0.68	its	0.26	's	0.75	his	0.36
mind	0.66	its	0.26	their	0.71	your	0.37	enormous	0.67	another	0.25	to	0.74	your	0.34
husband	0.65	our	0.26	'	0.71	its	0.37	opportunity	0.67	her	0.25	its	0.73	's	0.33
matters	0.65	your	0.26	her	0.70	a	0.36	planned	0.67	any	0.25	similar	0.73	every	0.33
party	0.65	their	0.25	its	0.70	's	0.36	military	0.66	his	0.22	and	0.73	its	0.33

syntactically similar (possessive)

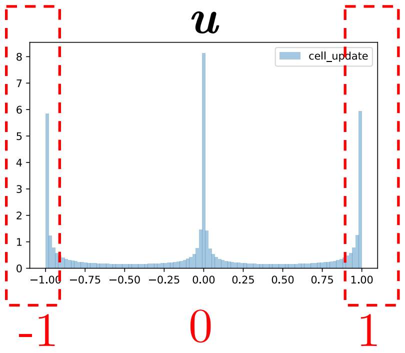
co-occurrence words

list of similar words : understanding roles of internal vectors

- comparison of similar words to "her" using $\boldsymbol{h}$, $\boldsymbol{c}$, $\boldsymbol{u}$, and $\theta(\boldsymbol{u})$ vectors.
- In $\boldsymbol{h}$, both types of words that have similar meaning or syntactic function are gathered.
- In $\boldsymbol{u}$, words that have similar syntactic functions are gathered most well.
- Quantizing $\boldsymbol{u}$ to {-1,0,1} ($\theta(\boldsymbol{u})$) doesn't change the result so much.

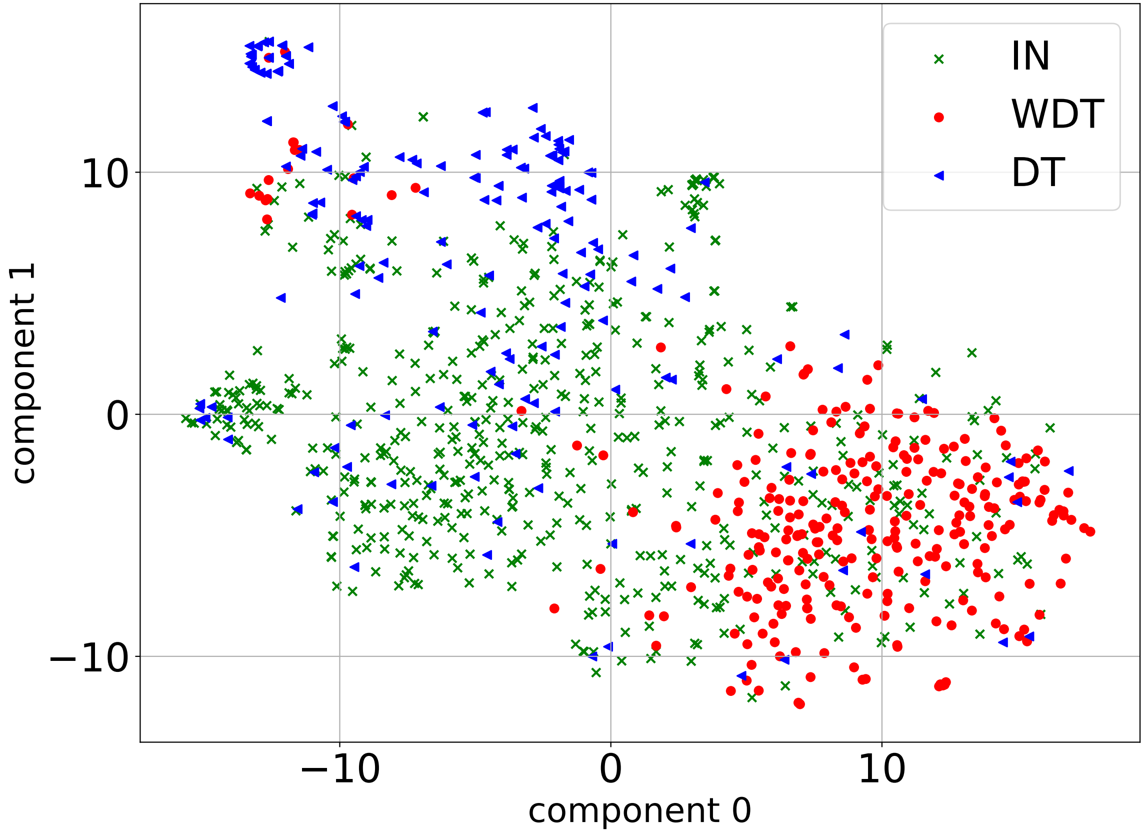
$\boldsymbol{h}$	$\boldsymbol{c}$	$\boldsymbol{u}$	$\theta(\boldsymbol{u})$
my	his	his	his
his	mother	my	my
mother	playing	the	the
husband	mind	its	its
mind	husband	our	your
wife	matters	your	their
their	party	their	's

Table 1: Most similar words with "her", based on different internal vectors in LSTM. $\theta()$ is a discretization by thresholds ± 0.9.



How the syntactic functions of the word "that" are embedded in u ?

- Word “that” is a representative ambiguous functional word.
- Different meanings are clustered although they are not completely separated.



IN: Preposition or subordinating conjunction, e.g. "if"
DT: determiner, e.g. "this"
WDT: Wh-determiner. e.g. "which"

Conclusion

Statistics of internal vectors ($\boldsymbol{c}$,$\boldsymbol{h}$,$\boldsymbol{u}$,$\boldsymbol{f}$) :

- Characteristic semi-quantization is observed for every internal vector.

Analyses of cell-state vector ($\boldsymbol{c}$):

- For Paren data, in $\boldsymbol{c}$, there is a completely quantized element that acts as a counter of the nesting depth of the parentheses.
- For Tag data, a linear sum of the elements of $\boldsymbol{c}$ can act as a counter of the nesting depth.

Analyses of cell-update vector ($\boldsymbol{u}$):

- POS is best represented in cell-update vector $\boldsymbol{u}$.
- Syntactic functions the word "that" has can be clustered in $\boldsymbol{u}$.

Related work about capability of LSTM-LMs w.r.t. capturing syntactic information

Empirical analyses :

- Synthetic data
  - Dyck-1,2 and shuffle of Dyck-1 languages (Suzugun et al. 2019)
  - SP-k languages (Enes et al. 2017)
  - Early studies for LSTMs with few dimensions (Prez-Ortiz et al., 2003; Schmidhuber, 2015)
- Real data
  - a lot of studies

  e.g. using number agreement to check if it captures syntax when viewed from the prediction result.(Linzen et al. 2016)

Theoretical analyses :

- expression capabilities are investigated: relation to counter machines are found. (Weiss et al. 2018, Merrill 2019)

Thank you for your listening.