LSTM (Long Short-Term Memory) の内部を覗く

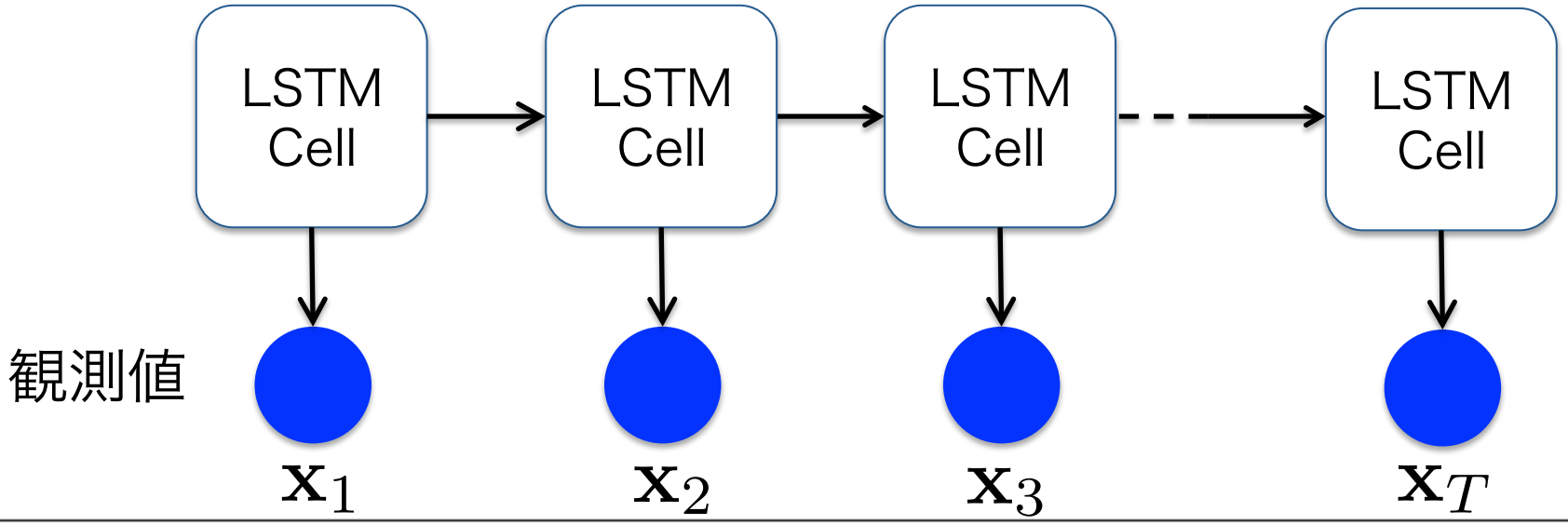
持橋大地
数理・推論研究系 学習推論グループ
daichi@ism.ac.jp

統計数理セミナー 2019-5-15 (水)

※ 本研究は、柴田千尋氏(東京工科大)・内海慶氏(デンソーアイティーラボラトリ)との共同研究です.

LSTMとは

- LSTM (Long Short-Time Memory) (Hochreiter+ 97) … 再帰的ニューラルネットの一種
- 時系列データをモデル化するために、様々な分野で頻繁に使われている
- 非線形HMMの一種



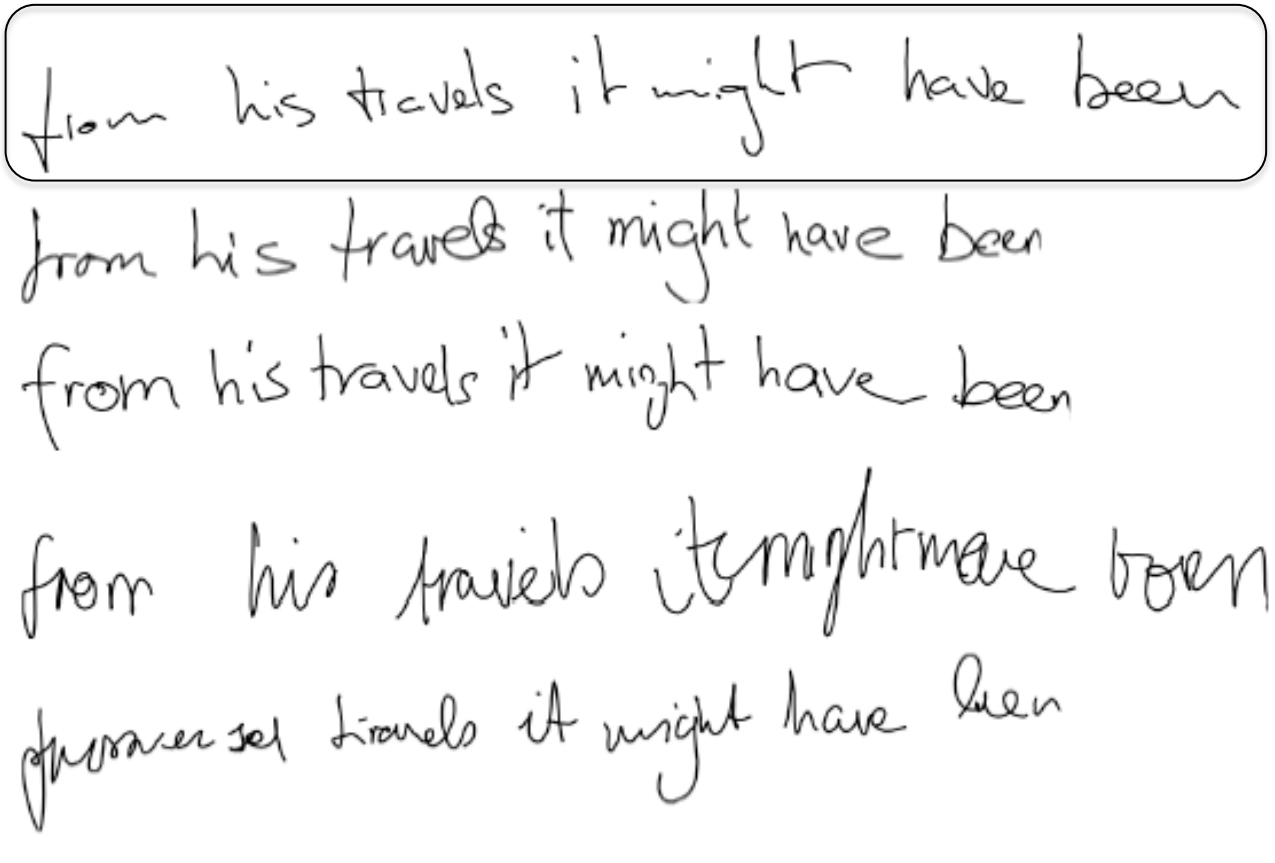
LSTMからの生成 (1)

- ランダムに生成されたWikipedia (Graves+13)

```
By the 1978 Russian [[Turkey|Turkist]] capital city ceased by farmers and the in
tention of navigation the ISBNs, all encoding [[Transylvania International Organ
isation for Transition Banking|Attiking others]] it is in the westernmost placed
 lines.  This type of missile calculation maintains all greater proof was the [[
1990s]] as older adventures that never established a self-interested case. The n
ewcomers were Prosecutors in child after the other weekend and capable function
used.

Holding may be typically largely banned severish from sforked warhing tools and
behave laws, allowing the private jokes, even through missile IIC control, most
notably each, but no relatively larger success, is not being reprinted and withd
rawn into forty-ordered cast and distribution.

Besides these markets (notably a son of humor).

Sometimes more or only lowed &quot;80&quot; to force a suit for http://news.bbc.
co.uk/1/sid9kcid/web/9960219.html ''[[#10:82-14]]''.
&lt;blockquote&gt;

===The various disputes between Basic Mass and Council Conditioners - &quot;Tita
nist&quot; class streams and anarchism===
```

LSTMからの生成例 (2)

- 手書き文字の生成 (Graves+13)

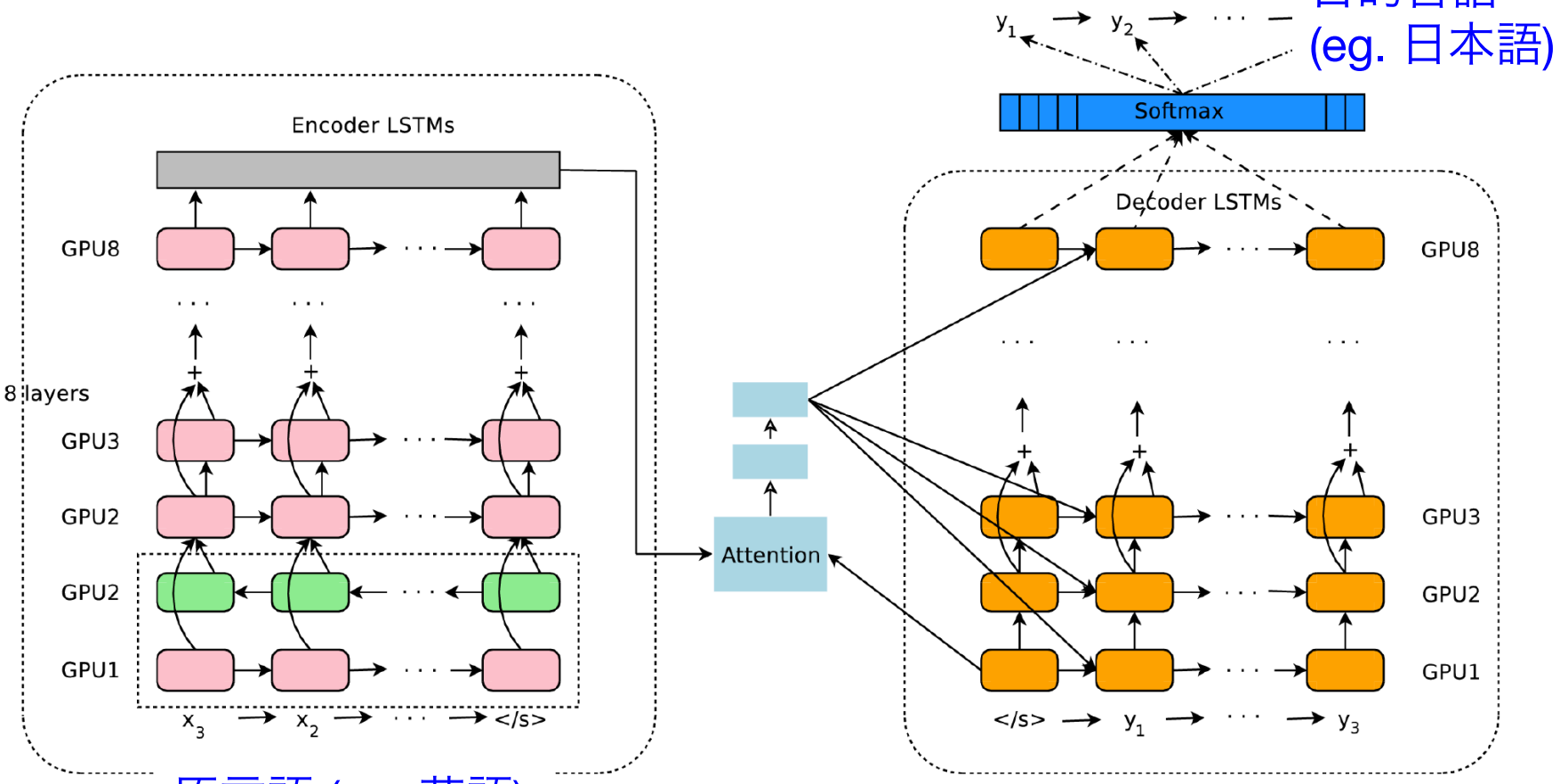


LSTMの使用

- 自然言語処理では、あらゆるタスクに使われている
  - LSTMなしでは、研究が成り立たないほど
  - 前向きLSTMと後向きLSTMを同時に動かし、隠れ状態をconcatしたBidirectional LSTMも多い
- ニューラル機械翻訳 (Google NMT, Wu+ 2016) はLSTMを8層結合したもの
- Neural Turing Machine (Graves+ 2014) などにおいては、コントローラをLSTMで動かしている

ニューラル機械翻訳

- Googleニューラル機械翻訳: 8層の双方向LSTM＋アテンションのエンコーダ/デコーダ



LSTMの動作について

- ほとんど解明されていない！
- いくつか研究があるが (Karpathy+16)(Ayache+18)(Schmidhuber+15)、
  すべてブラックボックスとしてLSTMを扱い、何ができるかを調べるもの

LSTMの中身

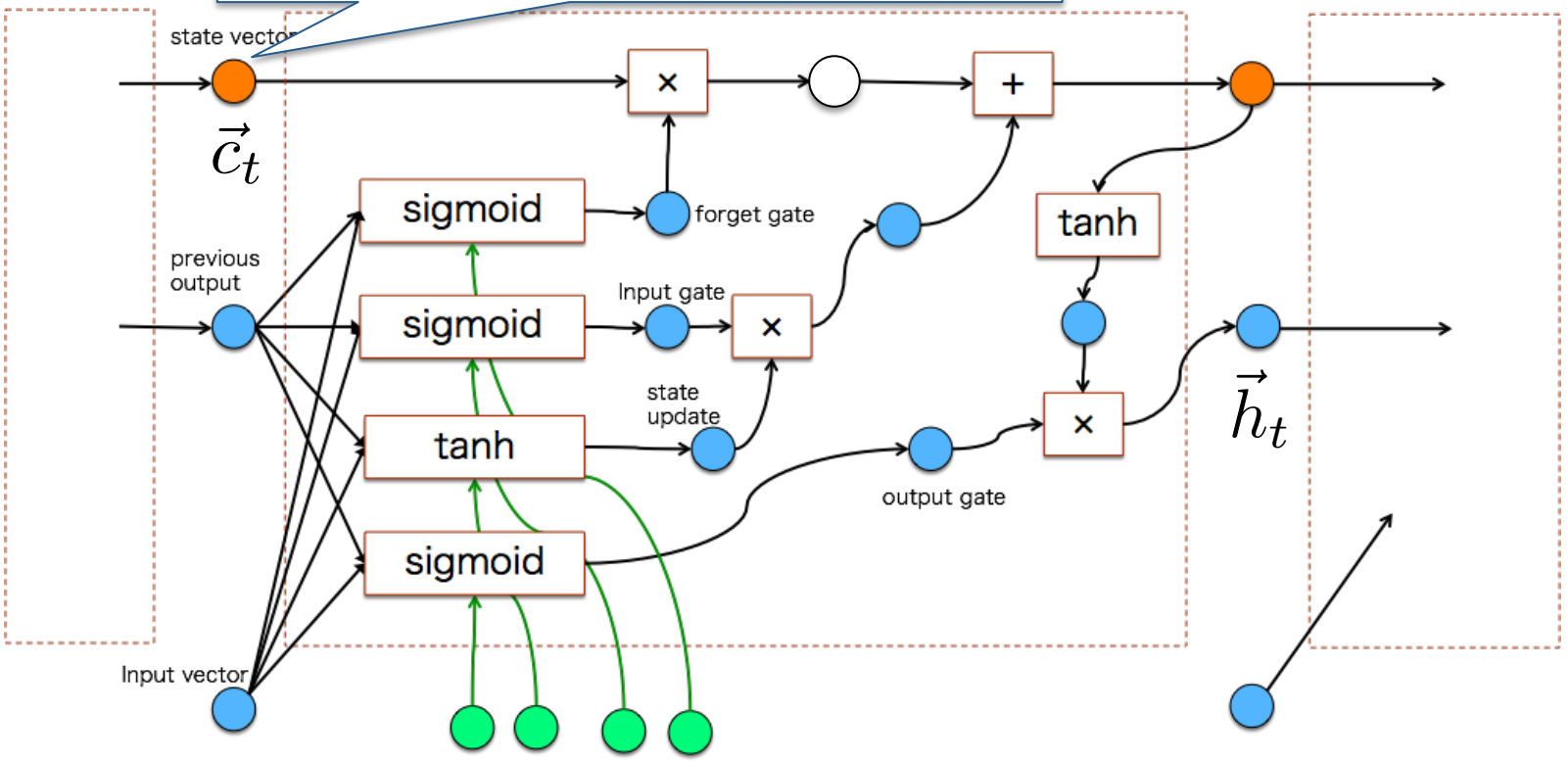
- LSTMセルの構造

$\vec{c}_t$ の更新則

$$\vec{c}_{t+1} = \vec{f} \odot \vec{c}_t + \vec{c}_{\mathrm{up}}$$

$\vec{f}$: $[0,1]^D$　　$\vec{c}_{\mathrm{up}}$: $[-1,1]^D$

文脈ベクトル $\vec{c}_t$ が受け継がれる



数式で表すと

$$(\vec{h}_{t+1}, \vec{c}_{t+1}) = \mathrm{LSTM}\,(\vec{h}_t, \vec{c}_t, \vec{w}_t, \Theta)$$

$$\vec{c}_{t+1} = \vec{f} \odot \vec{c}_t + \vec{c}_{\mathrm{up}}$$

$$p(w_{t+1}|w_1 \cdots w_t) = s(W\vec{h}_{t+1} + b)$$

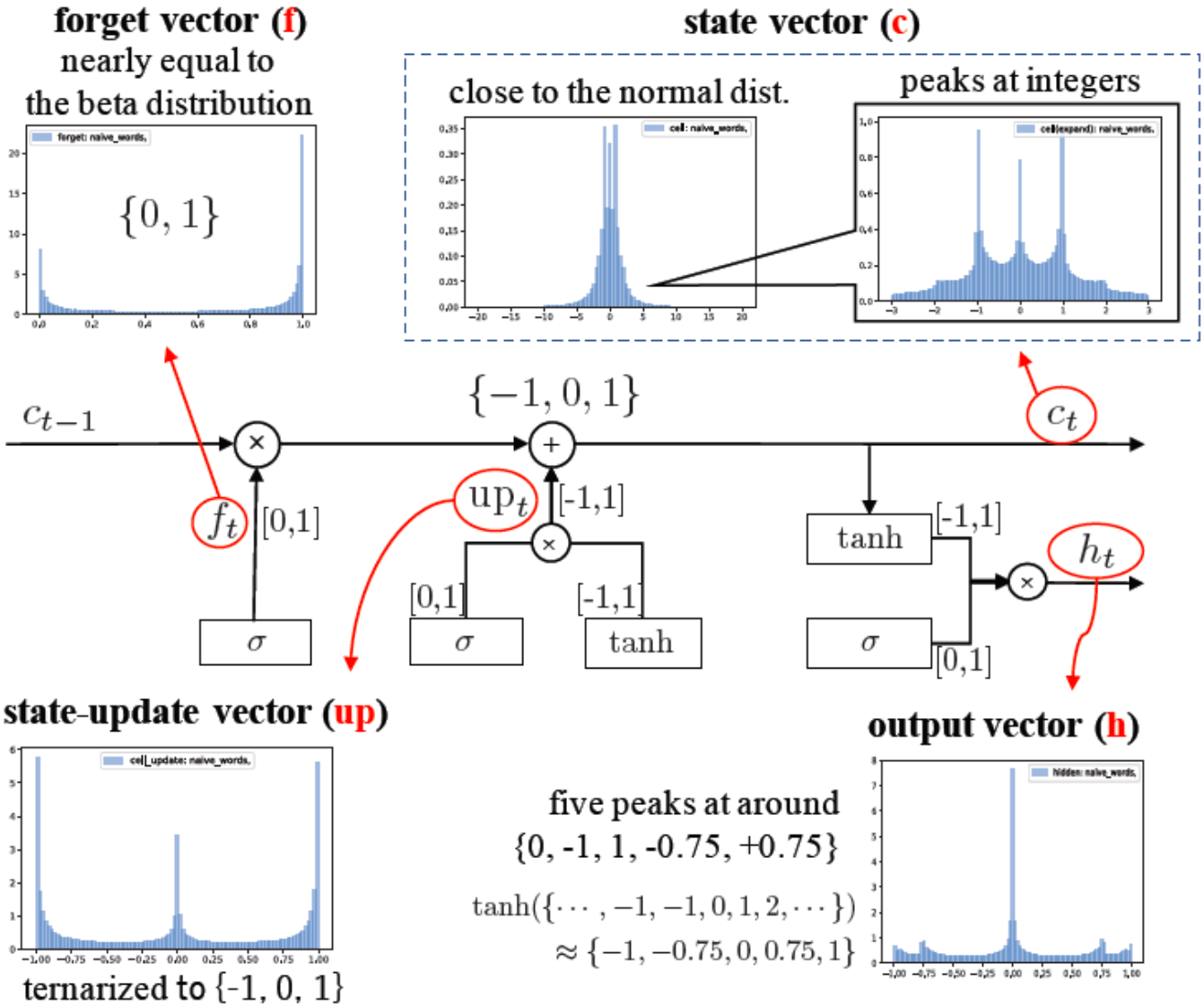
⊙はアダマール積

- sはソフトマックス(多値ロジスティック)関数

$$s(\vec{x}) = \left(\frac{e^{x_1}}{\sum_k e^{x_k}}, \frac{e^{x_2}}{\sum_k e^{x_k}} \cdots, \frac{e^{x_K}}{\sum_k e^{x_k}} \right)$$

- 文脈ベクトル $\vec{c}_t$ と、そこへの更新 $\vec{c}_{\mathrm{up}}$ に長期状態がエンコードされている

内部ベクトルの分布の様子



- かなり離散的に近く分布している
- 忘却ベクトルはβ分布、c_tは正規分布＋離散的なピーク

分析に用いたデータ

- Penn Treebank, Wall Street Journal コーパス (Marcus+ 1994)
- WSJの新聞記事の構文解析済みデータ、49000文
  - S式 (Lisp形式) で構文が付与されている
  - 構文解析の研究に標準的に用いられるコーパス
- 予定：Emacs付属のLispプログラム群など

学習のタイプ

- 次の5つのタイプにデータを前処理して学習した

(1) 括弧のみ残す
例: ST ((( )) (( ) (( )))) ED

(2) タグ付き括弧

例: ST (S (NP (DT DT) (N N) NP) S) ED

(3) タグなし括弧＋単語

例: ST ((John) (will (visit (there))))) ED

(4) タグ付き括弧＋単語
例: ST (S (NN John NN) (VP will (V visit V) (N there N)..
(5) 生文

– 例: ST John will visit there ED

学習設定

- LSTM言語モデル、次の単語の予測確率を最大化
- 隠れ層 (hおよびc) の次元
  - 単語を含まない (1)(2) は100次元および200次元
  - 単語を含む (3)(4)(5) は1000次元
  - 語彙の総数は10000語程度
- Adam, Dropoutあり (確率0.2および0.5)

学習のタイプ

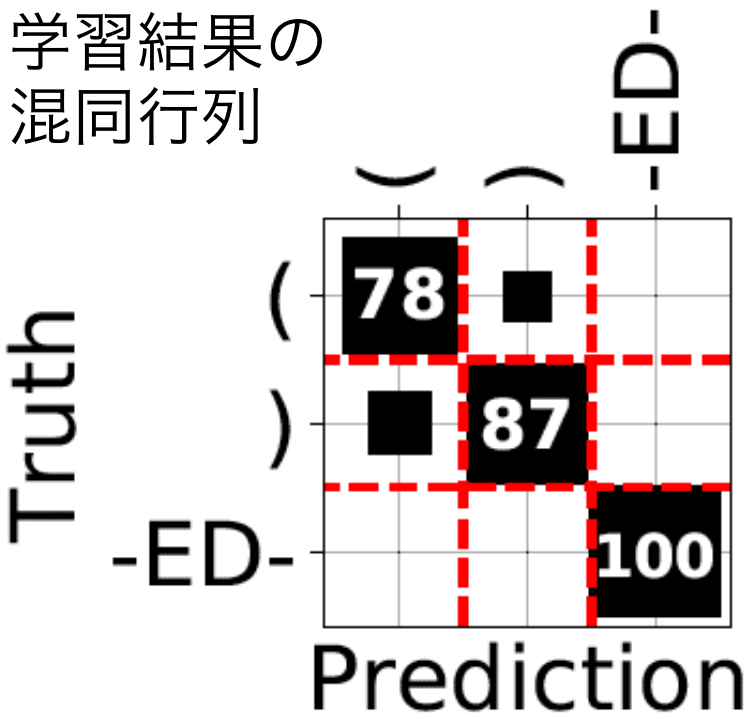
- 例えば、原データが
  (S (NP Pierre Vinken)
  (VP will (VP join
  (NP (DT the (NN board))))))　のとき、

(1)→ ST (( ) ((((( )))))) ED
(2)→ ST (S (NP NP) (VP (VP (NP (DT (NN NN) DT)
NP) VP) VP) S) ED
(3)→ ST ((Pierre Vinken) (will (join ((the (board
)))))) ED
(4)→ ST (S (NP Pierre Vinken NP)
(VP will (VP join (NP (DT the (NN board NN)
DT) NP) VP) VP) S) ED

学習タイプ(1)

- データの例: ST ((( )) (( ) (( )))) ED

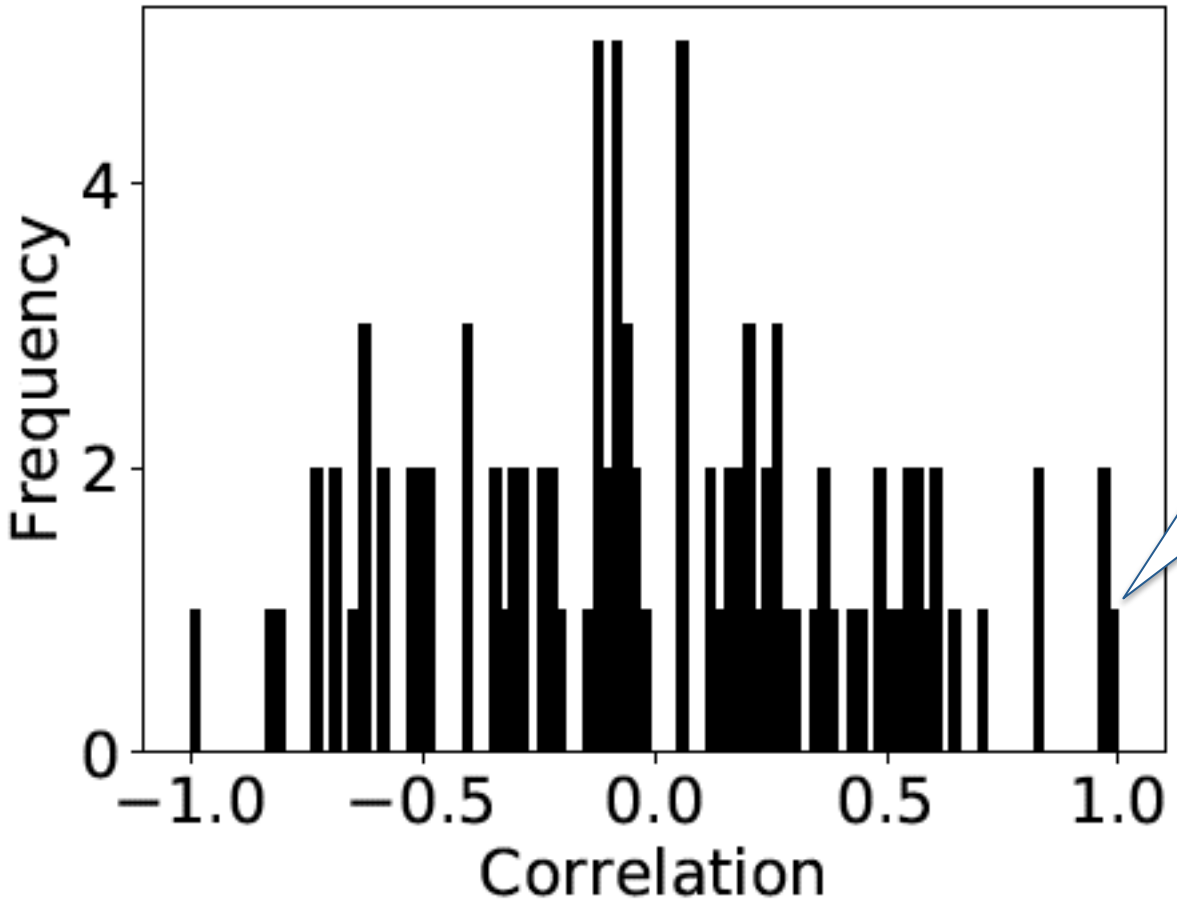
学習結果の
混同行列



- EDをほぼ100%の確率で予測できる
  → LSTMは括弧の深さを正確に数えている
- “(“, “)” もかなり正確に予測
  → もともと、”(“の次が“)”か”(“かは曖昧性がある (それでも高精度)

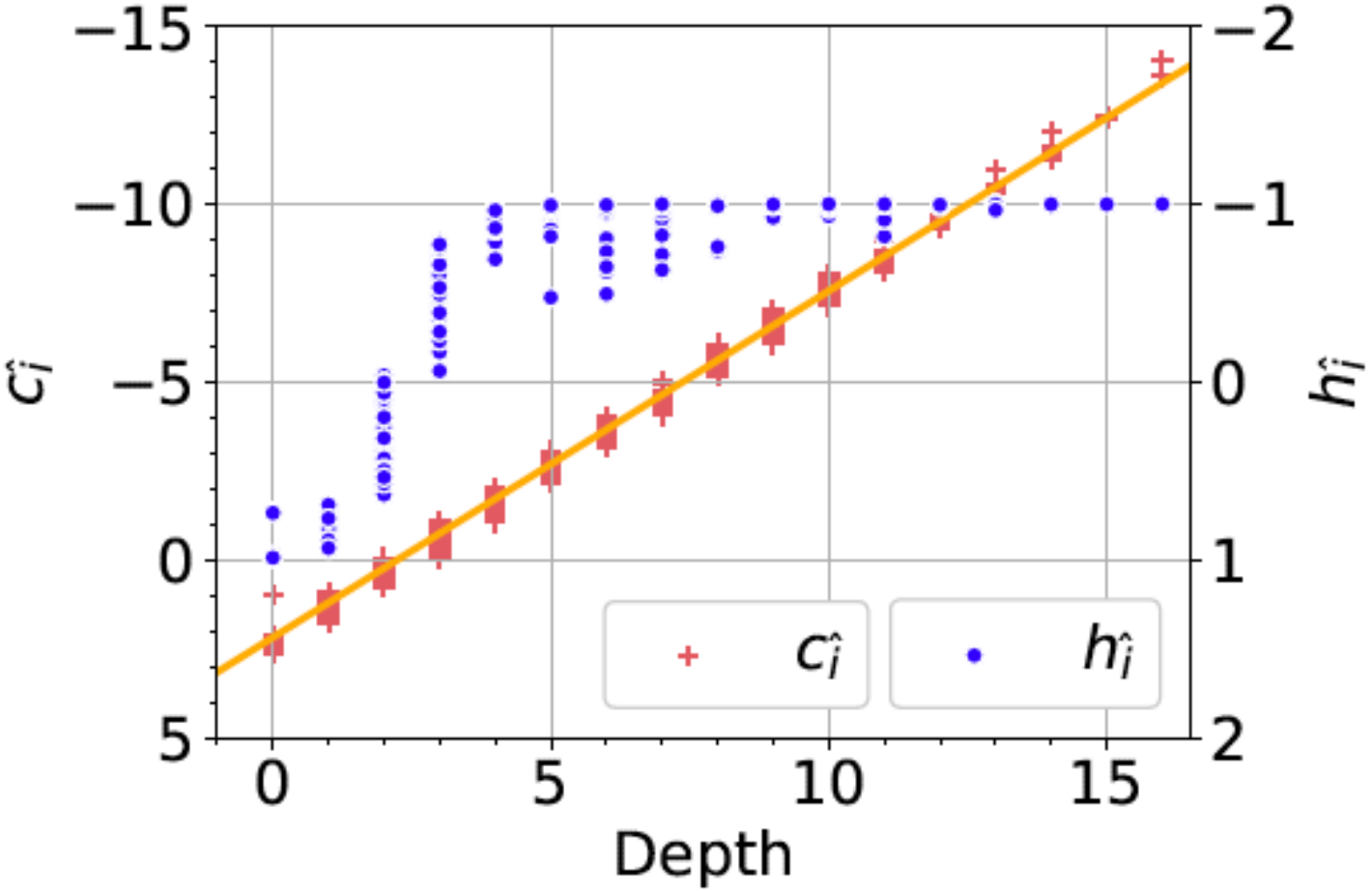
どの次元が構文の深さと相関しているのか?

- 括弧のネストの深さと、c_tの各次元との相関係数



非常に相関の高い次元が存在

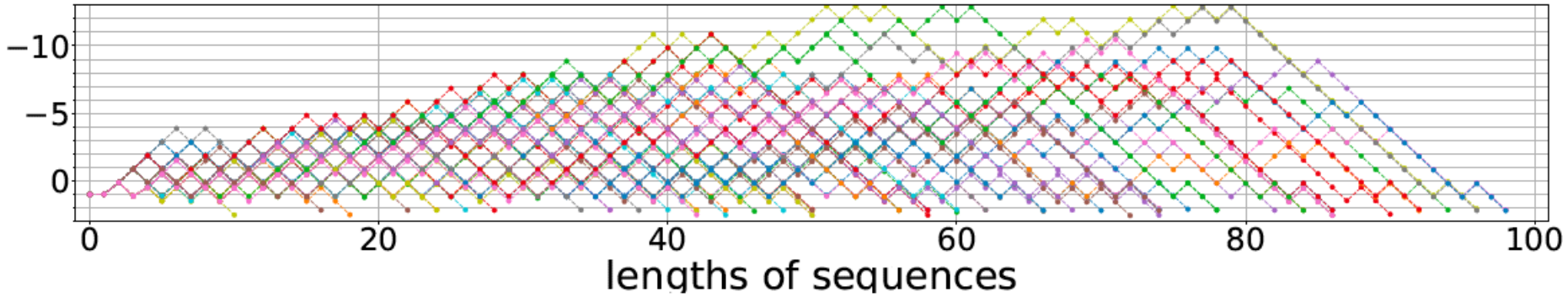
最も相関の高かった次元の値と構文の深さの関係



- h_t(青)よりも、c_t(赤)がきわめて高精度に相関する

文解析時のc_tの値

- 文解析時の、相関が最大だった次元のc_tの値の軌跡
  1つの色が1つの文
- 注意：c_tは連続値



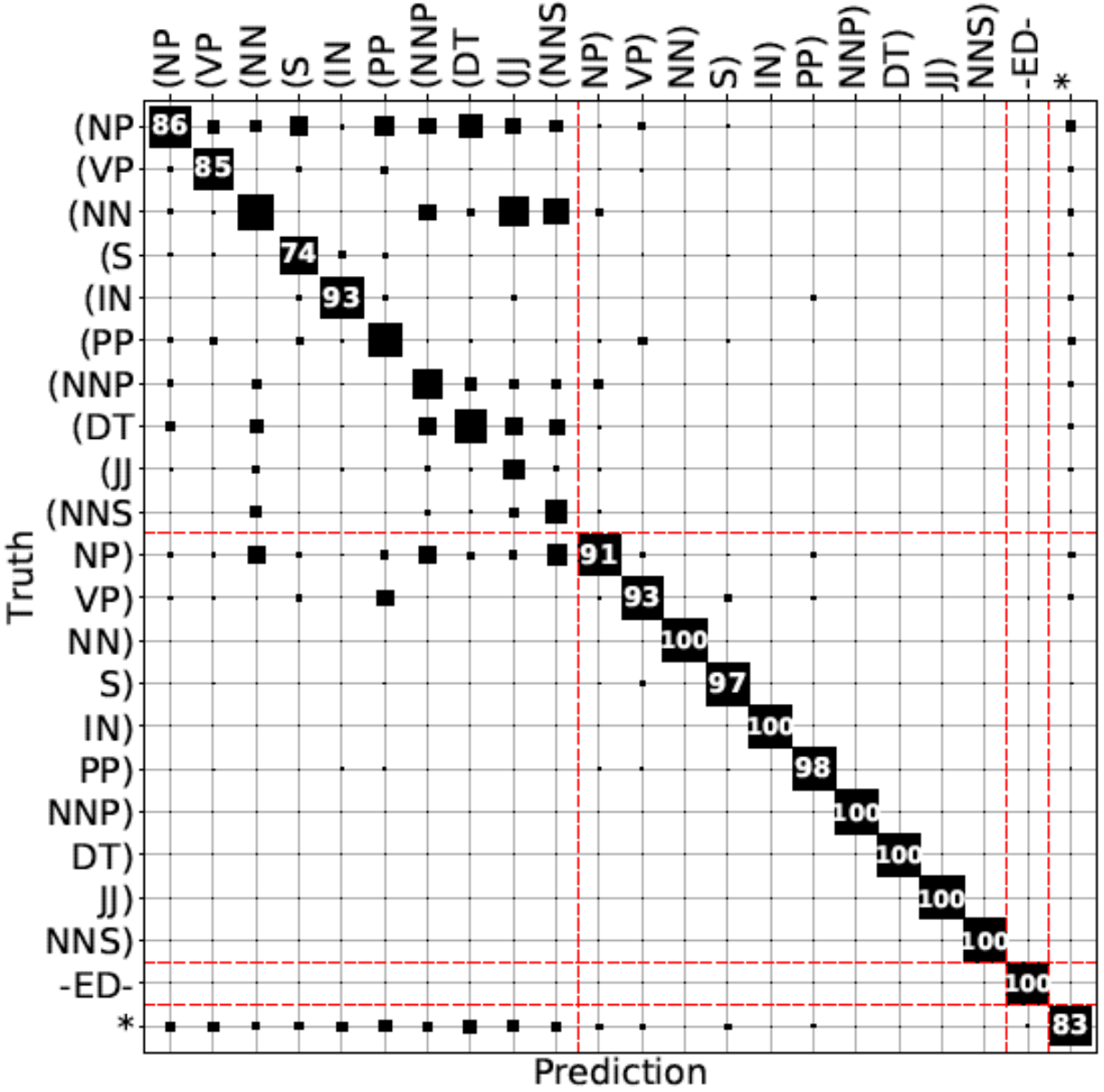
- ほとんどカウンターのように離散的にふるまっている！

学習タイプ (2)

- データの例:
  ST (S (NP NP) (VP (VP (NP (DT (NN NN) DT)
  NP) VP) VP) S) ED

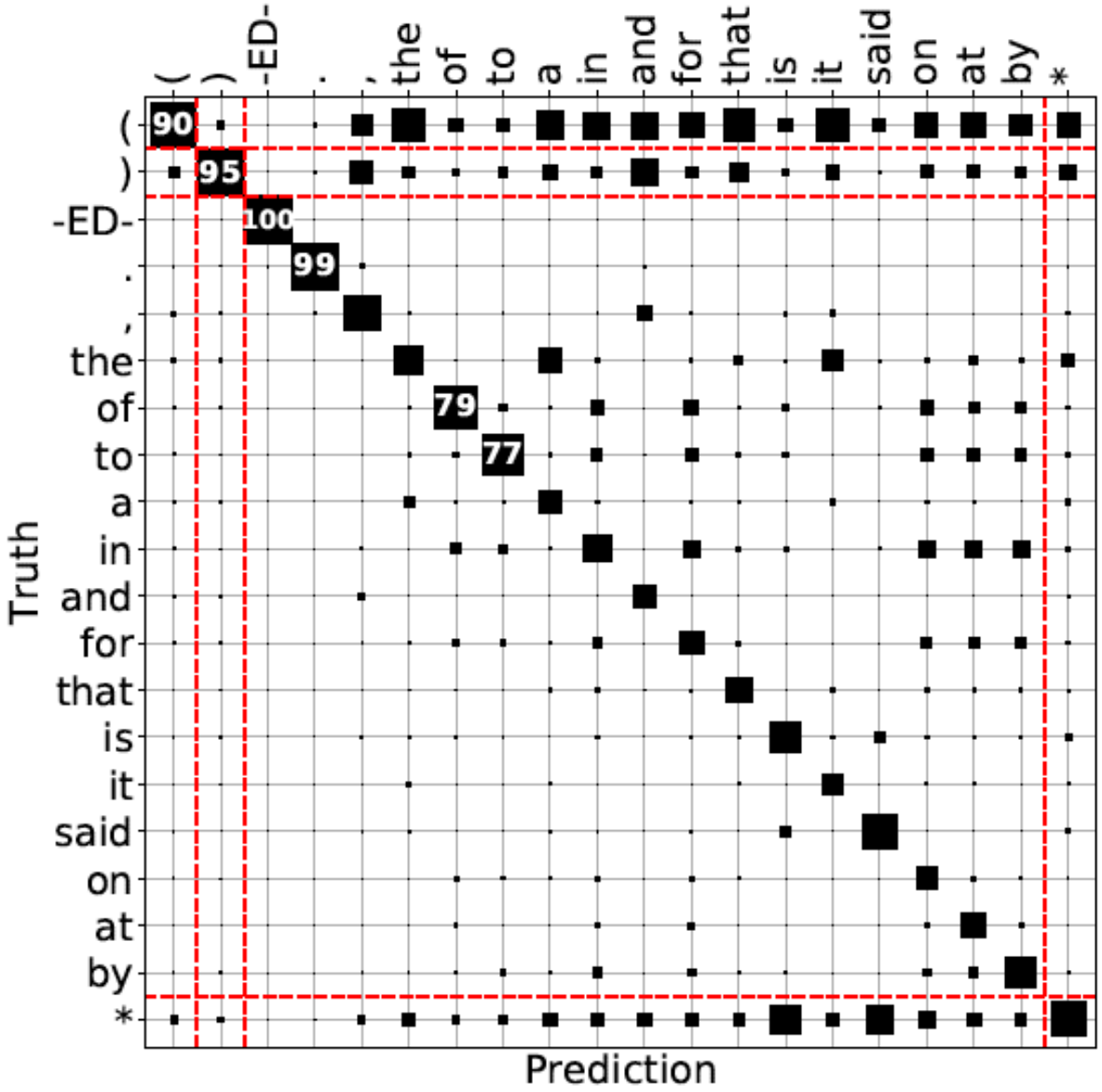
- 次の“単語”を予測する混同行列
- 注意: 学習には上の文字列は使っていないので、“(NP” と “NP)” が対応しているかは計算機には分からない
  - 実際には、IDにして“0 1 2 5 4 4 1 3 6 7 9 8 10 11” のような整数の系列をLSTMで学習

予測結果



- EDは100%正解→括弧を数えている
- “(XP”と“XP)”の対応を学習
- “(XP”より、“XP)”の予測が正確
→ 環境のネストを覚えている

学習タイプ (3)



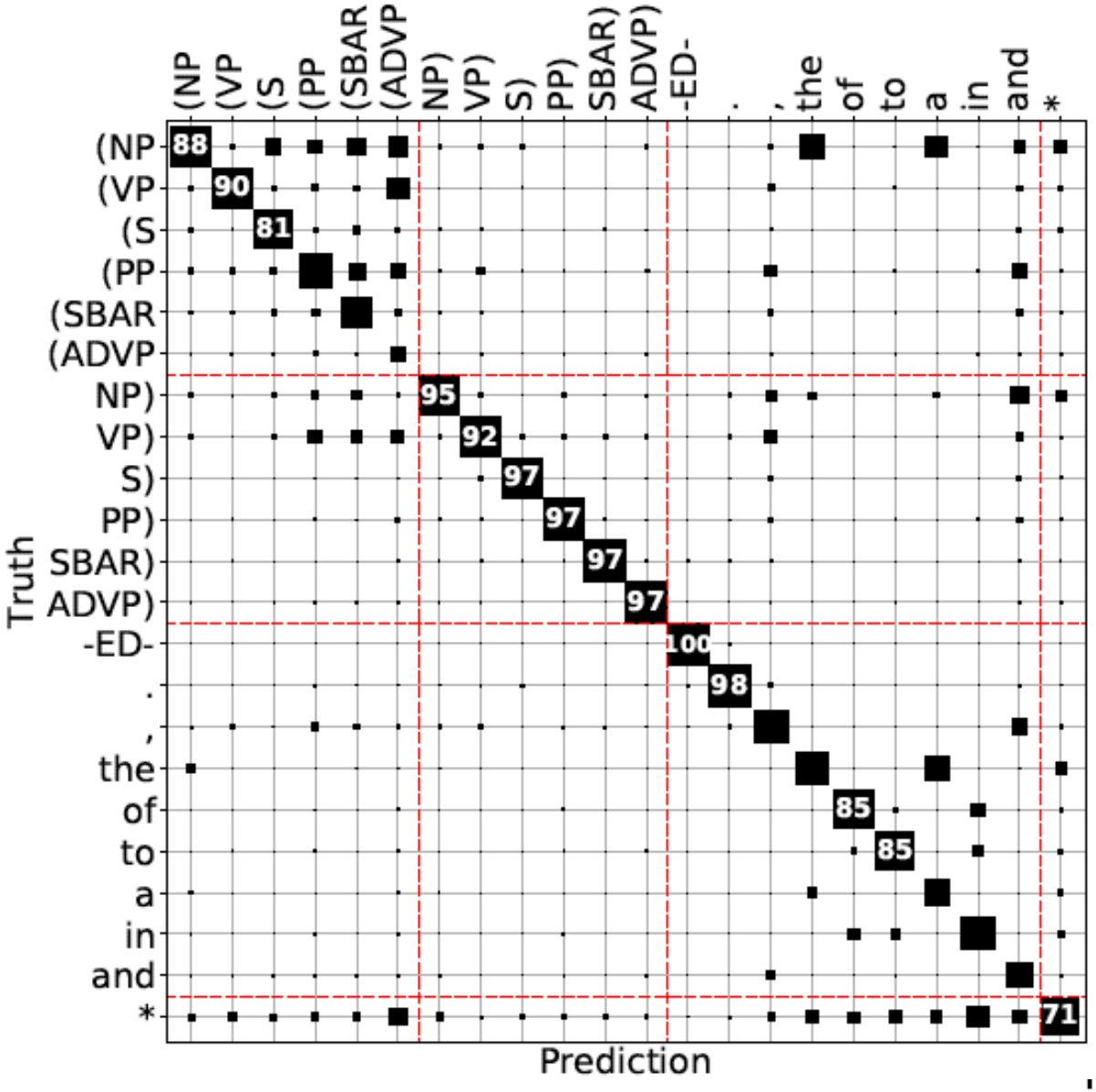
- 単語と ()
  だけ→やはり
  高精度に予測

学習タイプ (4)

- データの例:

  ST (S (NP Pierre Vinken NP)
  (VP will (VP join (NP (DT the (NN board NN)
  DT) NP) VP) VP) S) ED

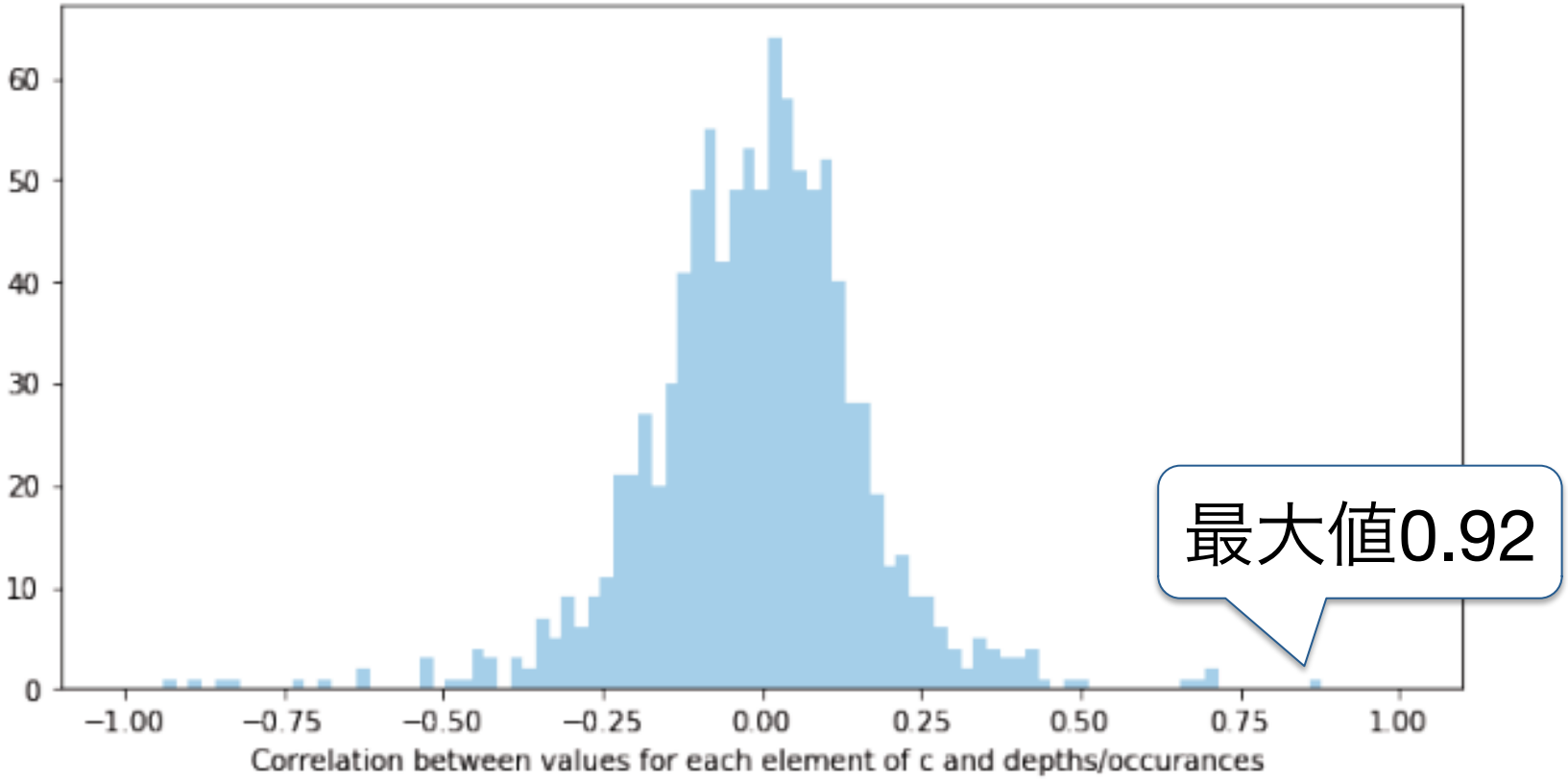
学習タイプ (4) : 混同行列



- 単語の情報と句の情報が加わることで、予測がより正確になった
- EDはやはり100%
  → 括弧の対応をすべて学習している

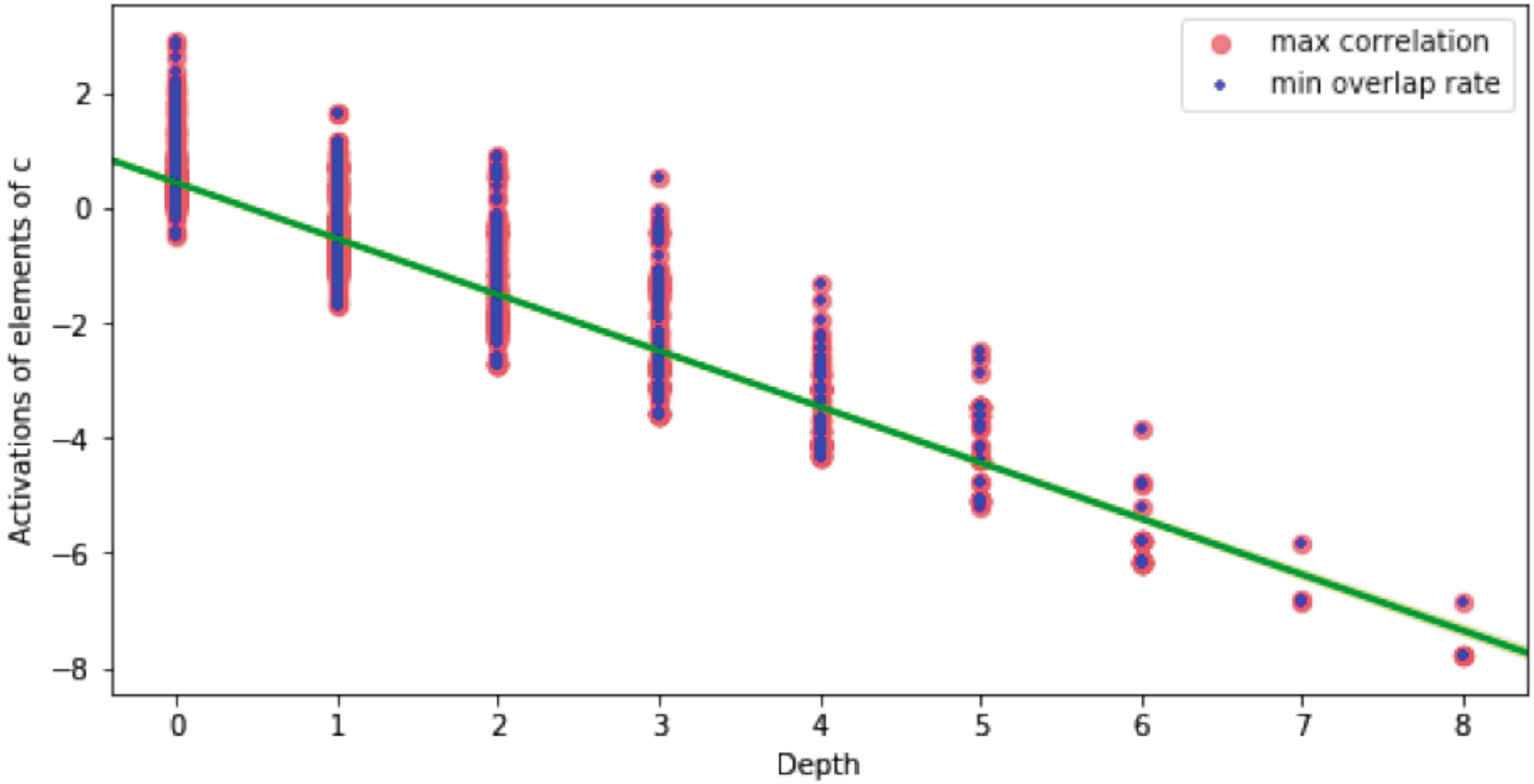
構文の深さと次元

- VP (動詞句) の深さと、1000次元のc_tの各次元との相関のプロット



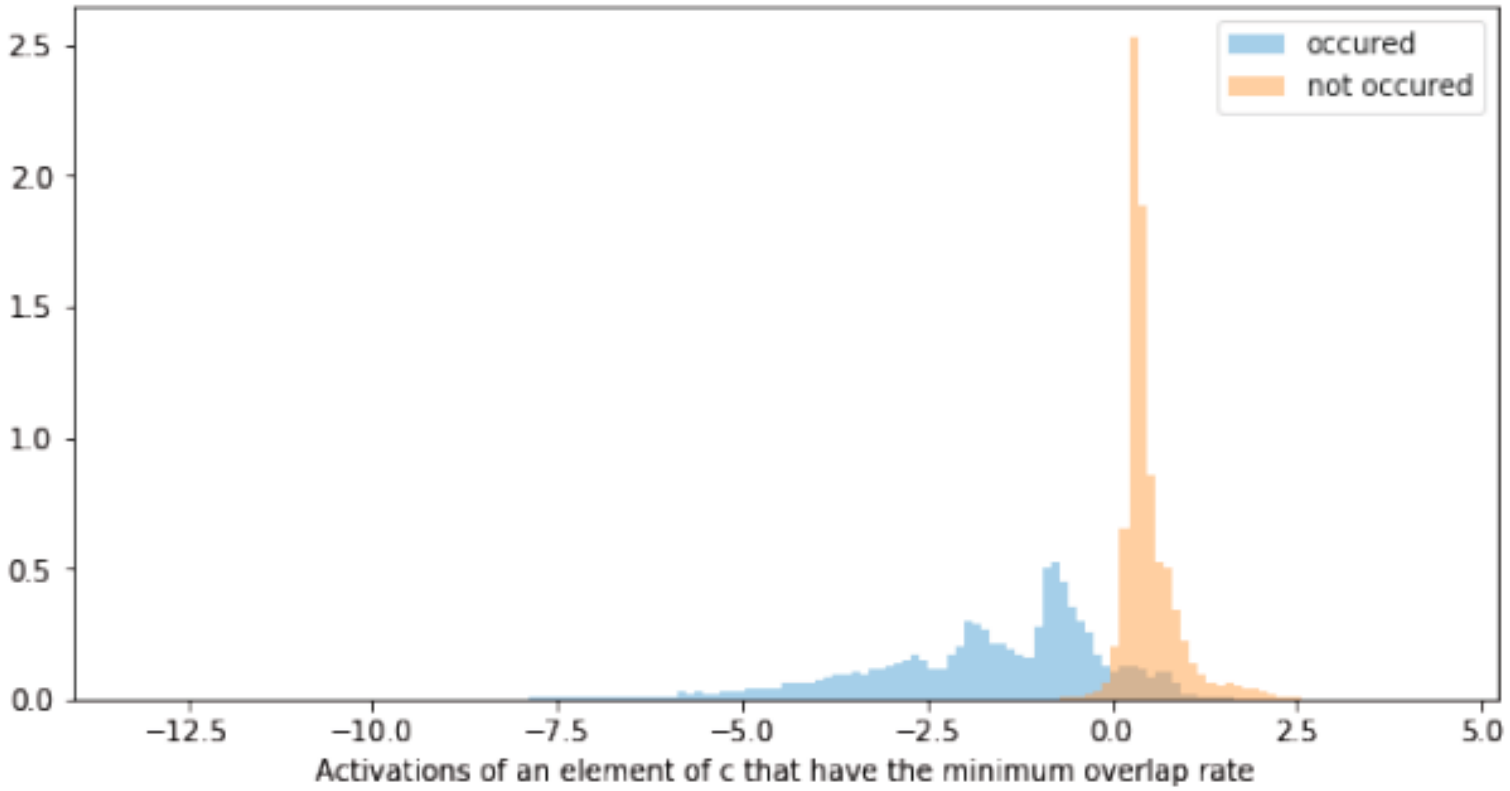
構文の深さと次元 (2)

- 相関最大の次元の値と、VPの深さとの関係



構文構造と次元

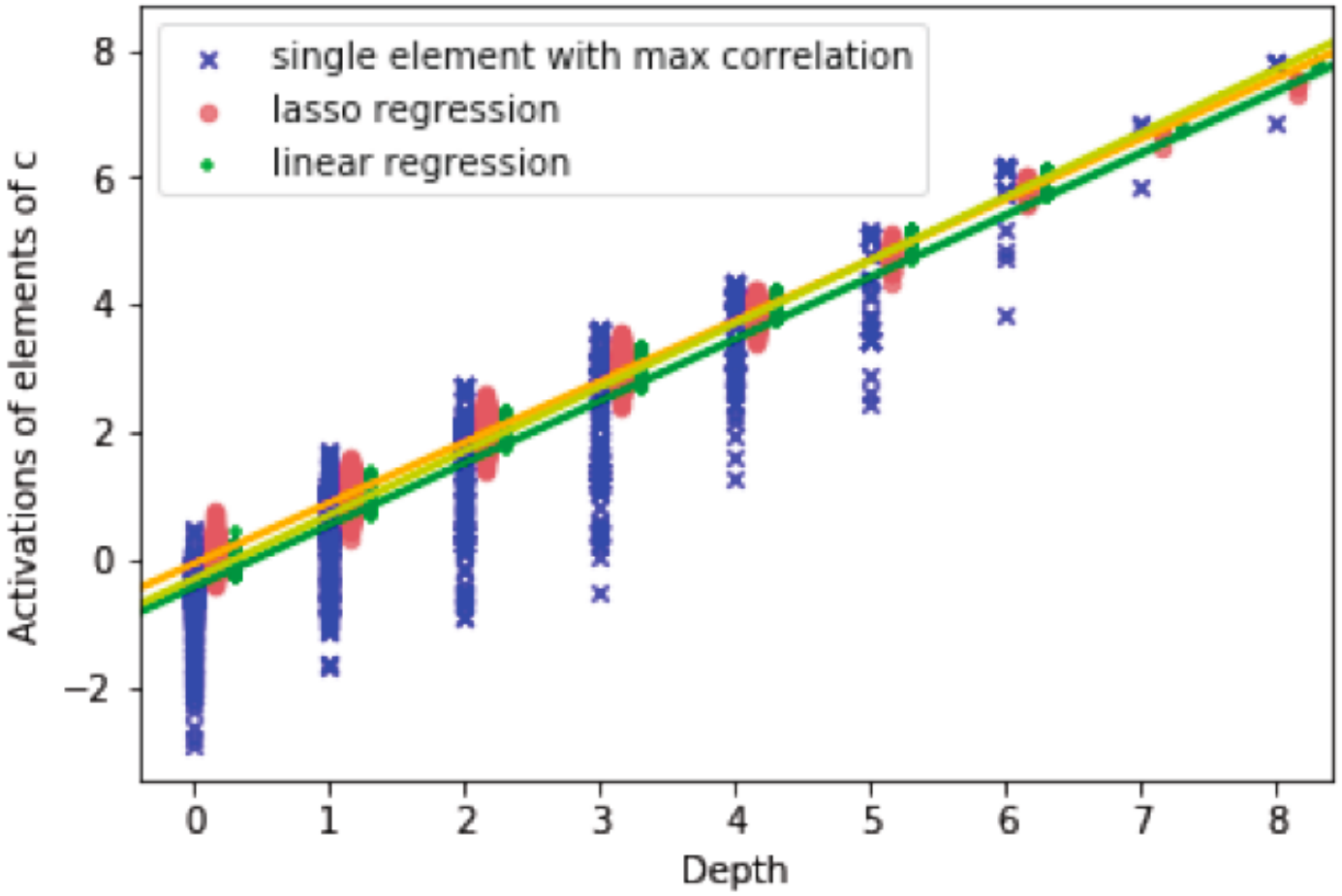
- ある次元の値だけで、VPの内外を判別できるか?
  → かなりできる (ヒストグラム重複率0.12)



構文構造とc_tの関係

- LSTMのモデルとしては、単一の次元で構文構造を予測するように学習しているわけではない
  → c_t からの回帰 (構文の深さ)
  → c_t からの識別モデル (VP, NPなどの内外判定)

c_t→構文の深さの回帰



● ●：線形回帰、●：Lasso回帰、×：単一の次元

VPの内外判定

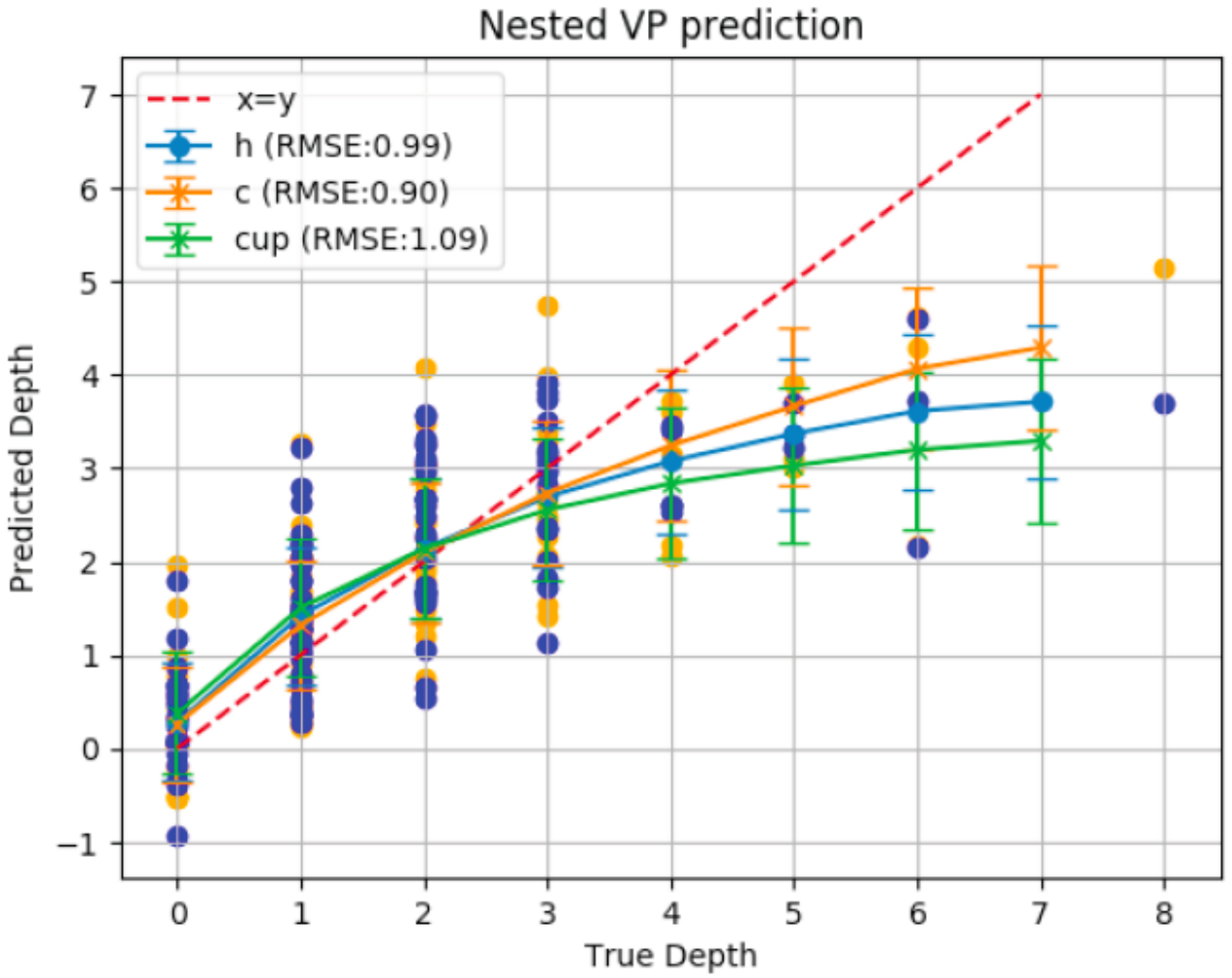
- c_tからのロジスティック回帰によるVPの内外の判定
- L1正則化により、関係する次元だけを残してスパースに

C	正解率	非ゼロ要素数 (1000 次元中)	非ゼロの割合
0.003	0.9996	134	13%
0.001	0.9992	100	10%
0.0003	0.998	71	7%
0.0001	0.991	51	5%
0.00003	0.97	27	2.7%
0.00001	0.87	12	1.2%

学習タイプ (5)

- 生文から学習、構文情報なし
- データの例:
  ST Pierre Vinken will join the board ED
  ST Kemper Financial Services Inc. , charging that program trading is ruining the stock market , cut off four big Wall Street firms from doing any of its stock - trading business ED

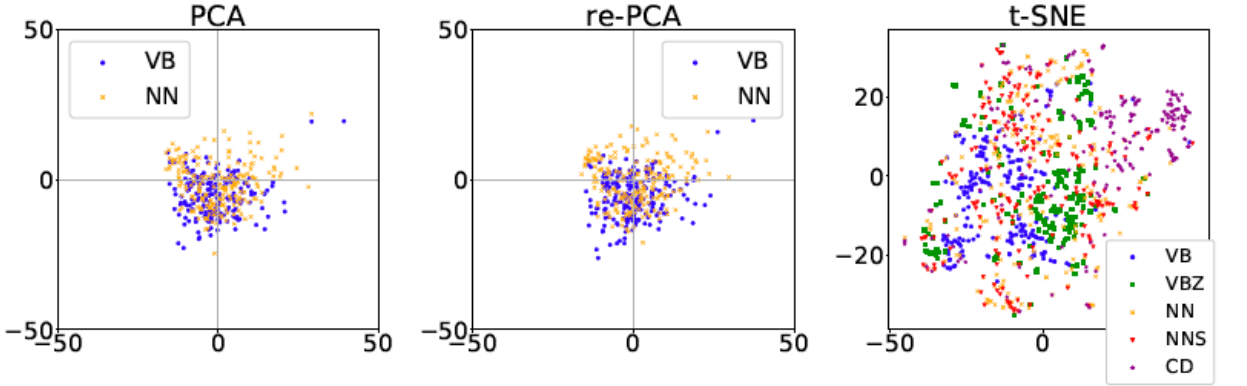
VPの深さの回帰



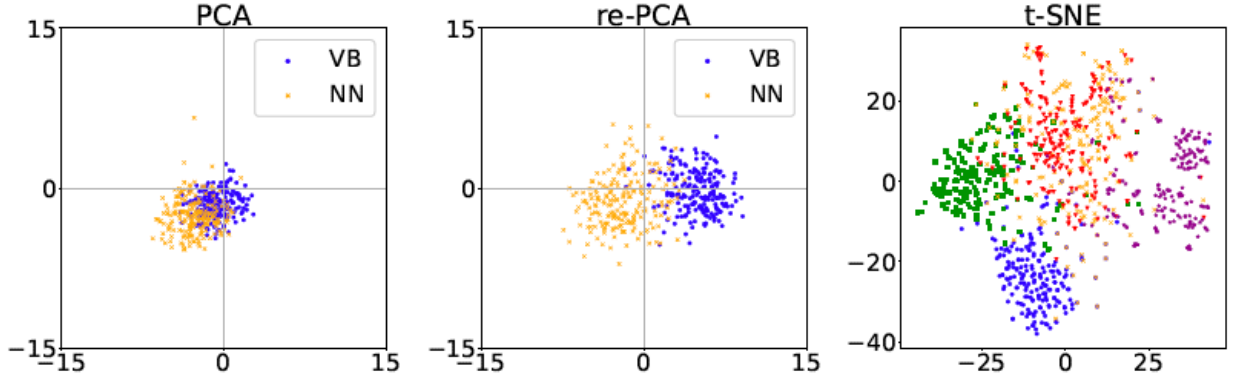
- 正解構文のVPの深さを予測
- 教師なしなので、関係は線形ではないが単調な関係
- cが相関を最もよく捉えている

品詞情報の学習

- どの情報が有用か? (VBとNNの判定)



cを使った場合

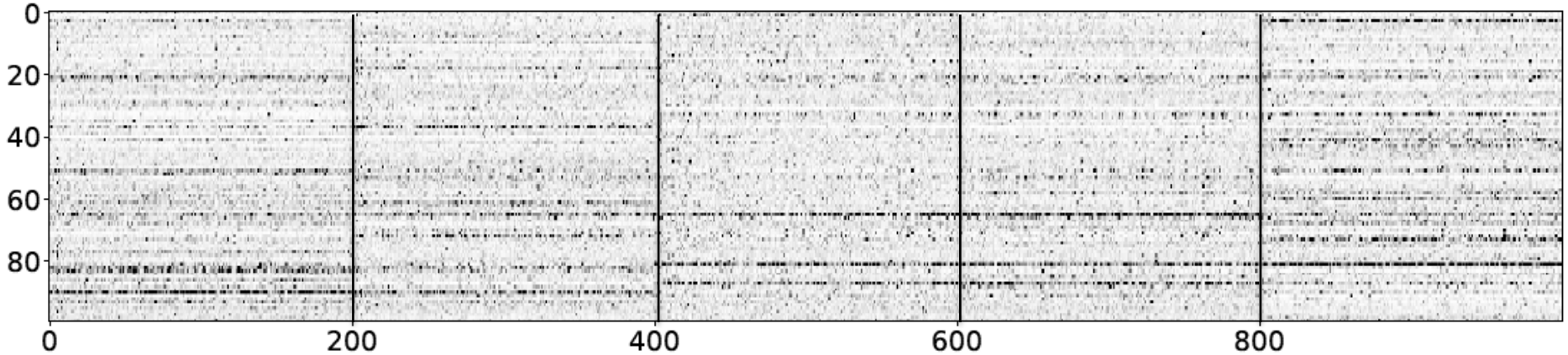


c_{up}を使った場合

- 文脈ベクトルcより、そこへの更新c_{up}がよい情報

品詞情報の学習 (2)

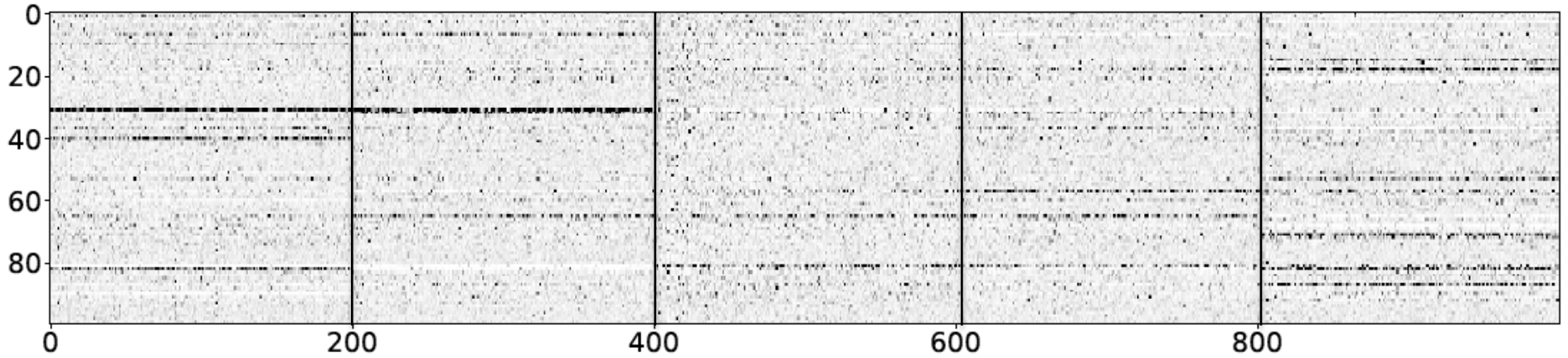
- 各品詞の単語に対応するc_{up}ベクトル(1000次元)をPCAで50次元に落としてみる
  - 負の値を可視化するため、同じデータに-をつけたものを後半にコピペして、expに乗せる (すべて正にする)



- 品詞は区別できるが、若干わかりにくい

品詞情報の学習 (3)

- 全データ(同じ単語が複数回登場)を一度PCAした後で、単語をすべて1回ずつに揃えてPCA (unique-PCA or re-PCA)

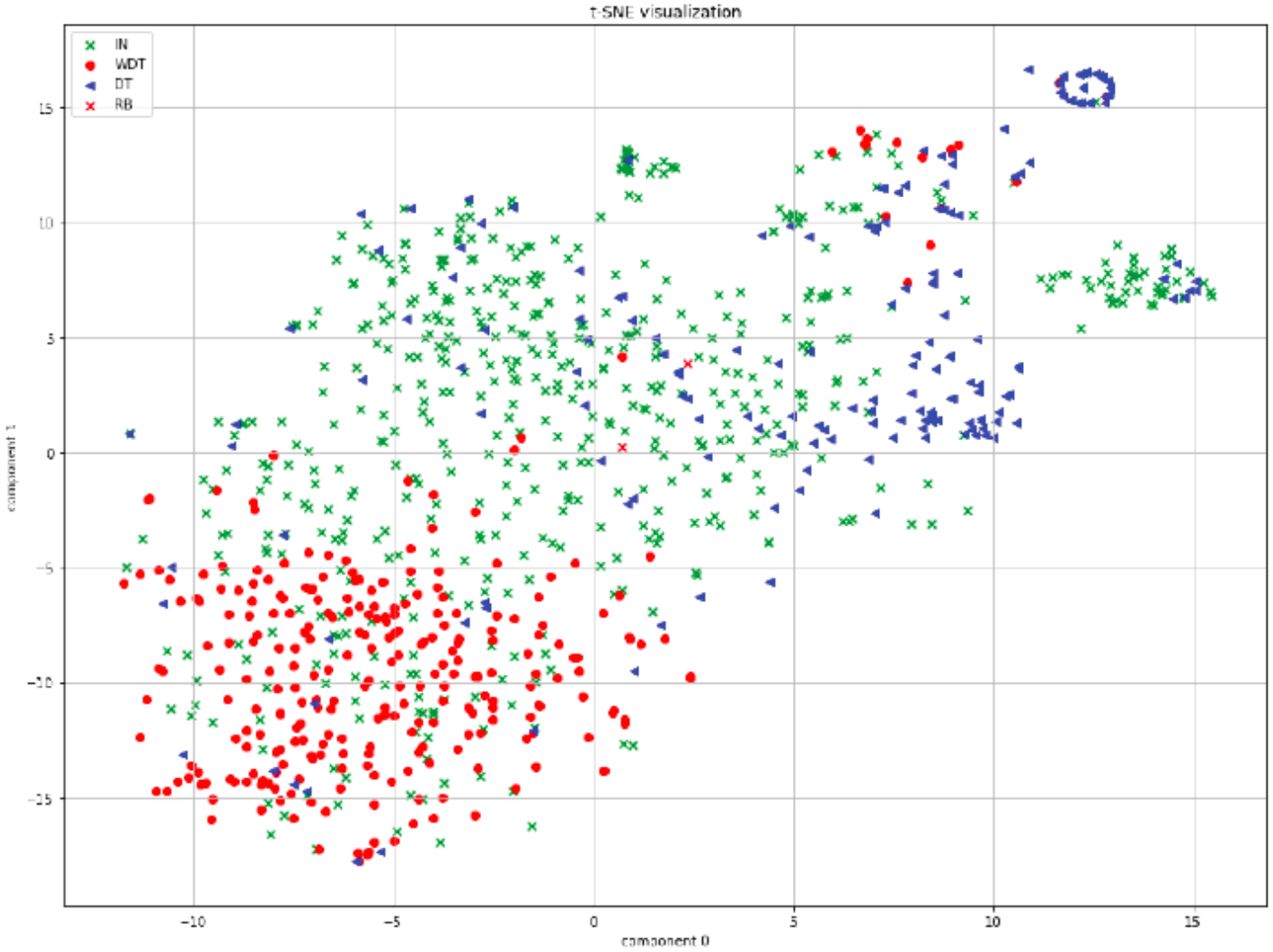


- 動詞の数の区別や、名詞の複数形も区別できた

品詞情報の学習 (4)

- 単語の文法的な働きは一意とは限らない
- “that” は、“this”, “if”, “which” とそれぞれ似た働きをする場合がある (他の例: given など)

品詞情報の学習 (5)



- これらはすべて“that”だが、色で表した品詞に従って分布している

まとめ

- LSTMは非常に頻繁に使われるモデルにもかかわらず、内部の動作が調べられていなかった
- 構文構造を与えたコーパスを使って、構文構造と内部ベクトルの関係を統計的に分析
  - 句構造のネストの深さは、特定の次元と高い相関
  - 内部ベクトルの少数の要素からの線形回帰で、構文の深さが非常に正確に予測できる
  - 上は、教師なし学習の場合でもほぼ同様に成り立つ
  - 文脈ベクトルの更新c_{up}が、単語の品詞とかなり対応している

今後の課題

- L_1正則化したLSTMを使って、LSTMを離散的に近く動作させ、文法と対応が取れるか調べる
- 自然言語以外の、ネストがあるデータで同様に動くか調べる (Emacs付属のLisp、scheme slib等)