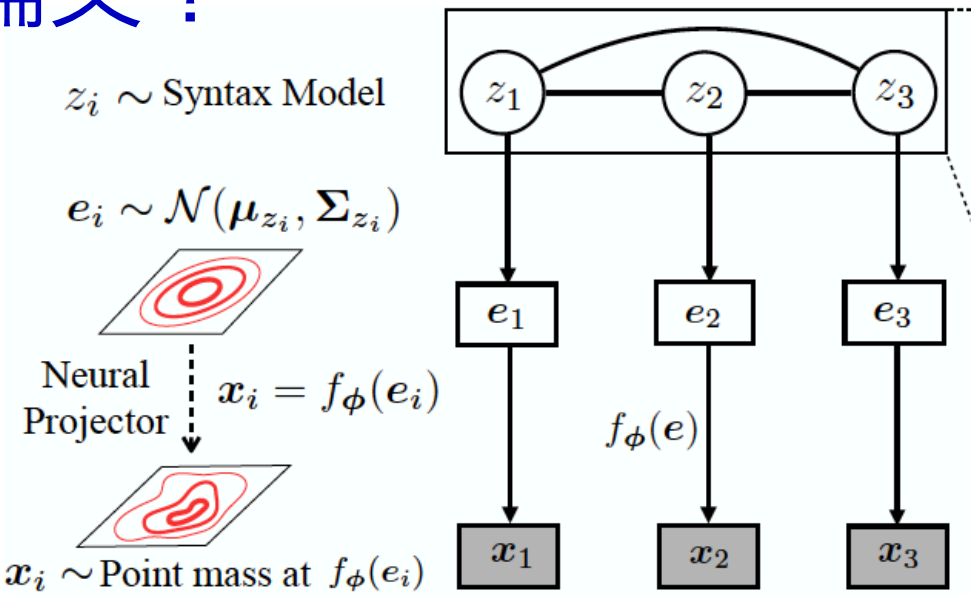
Unsupervised Learning of Syntactic Structure with Invertible Neural Projections

Junxian He, Graham Neubig, Taylor Berg-Kirkpatrick
EMNLP 2018

持橋大地
統計数理研究所
daichi@ism.ac.jp

最先端NLP 2019
2019-9-27 (金)

どんな論文？



- これまでのNLPの離散確率モデルで、単語ベクトルを扱えるようにする
- モデルが定義する「真の単語ベクトル」から学習済みの単語ベクトルへの写像をニューラルで学習
  - ヤコビアンが自動的に1になる可逆写像 (Dinh+ 2014)
  - 教師なし品詞学習、教師なし構文解析で最高性能

研究背景

- 自然言語処理では、今まで培われてきた統計モデルとニューラルネットの乖離が深刻
  - ニューラルモデルはそれだけで完結、複雑な構造を入れることができない
  - 確率モデルはニューラルと容易な接続が難しい
- ニューラル言語処理の中心は、単語ベクトル

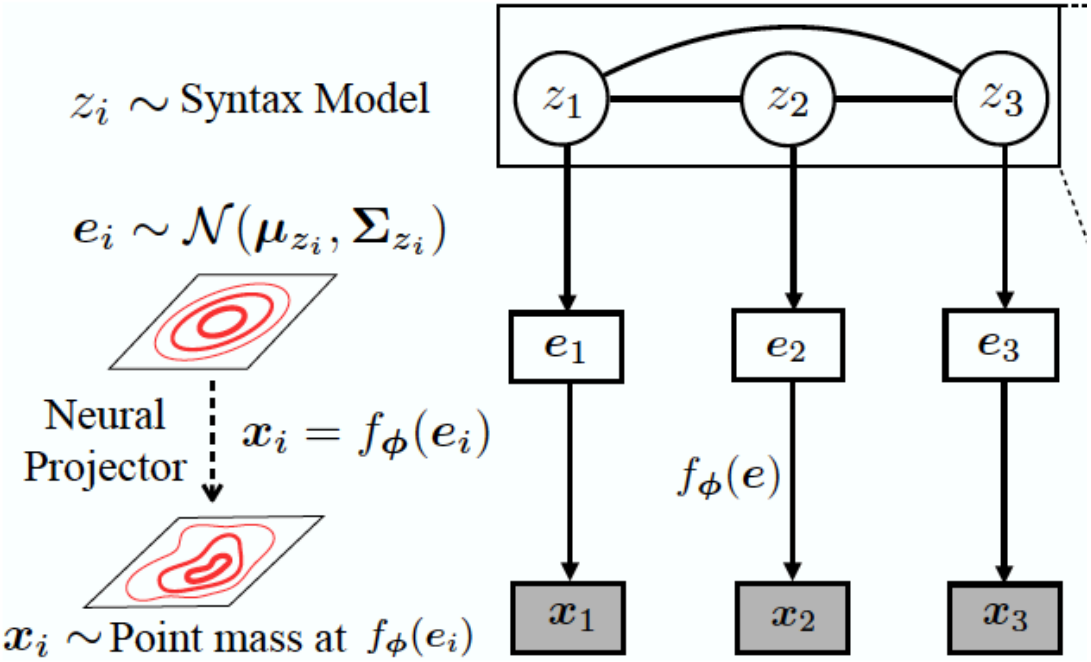
単語ベクトルの長所を統計モデルで生かすことができないか？

アドホックな方法

- Gaussian HMM (Lin+15), Gaussian LDA (Das+15)
- 単語ベクトルを「生成」？
- より重大な問題：
  観測値＝既存の単語ベクトルが、統計モデルから導かれるものと一致するとは限らない
  - 解いている問題に最適ではない
  - 単語ベクトル自体に、多様な作り方がある

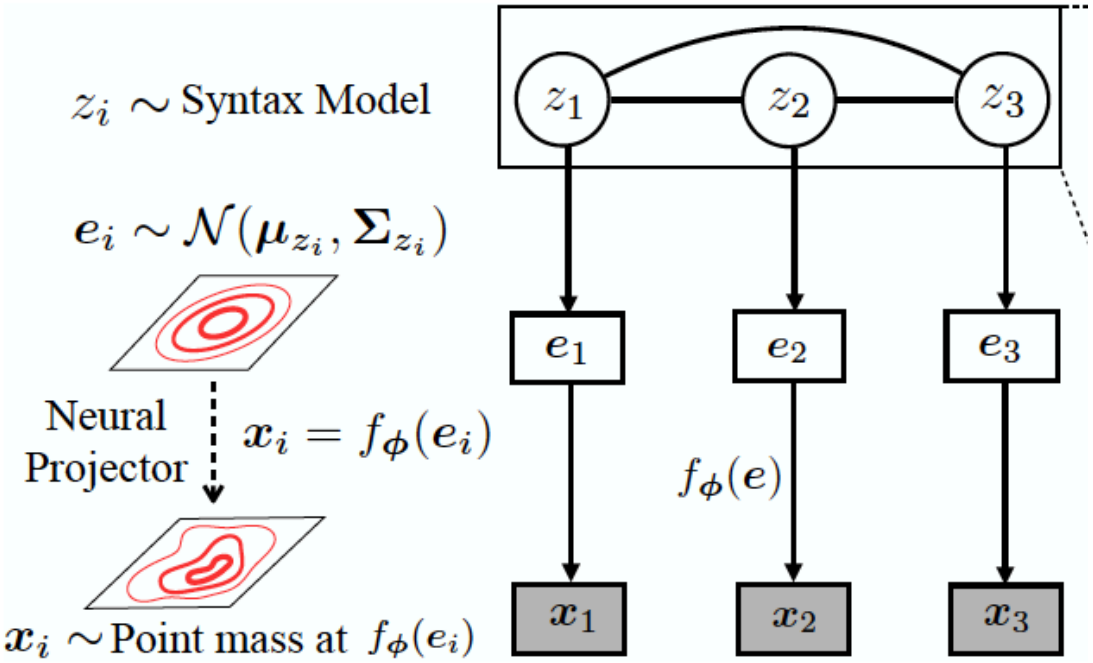
提案法

1. 確率モデルから、何らかの離散構造 z を生成
2. z から、“真の単語ベクトル” e を生成
3. e を関数 f で写像して、観測値つまり単語ベクトル $x = f(e)$ を生成



提案法 (具体例)

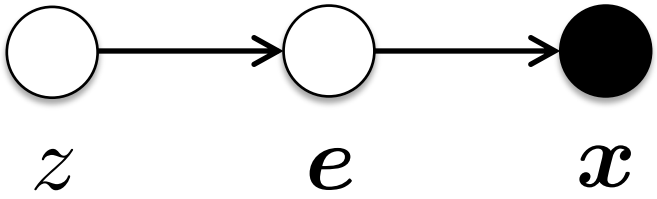
1. HMMから、状態ラベル z を生成
2. z ごとのガウス分布から、“真の単語ベクトル” e を生成 (各品詞が、ベクトル空間のガウス分布に対応)
3. e を関数 f で写像して、単語ベクトル x を観測



確率モデル：

$$\begin{cases} z_t \sim p(z_t|z_{t-1}) \\ e_t \sim \mathcal{N}(\mu_{z_t}, \Sigma_{z_t}) \\ x_t = f(e_t) \end{cases}$$

確率モデル



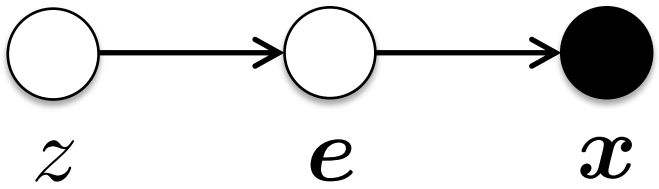
- $z \to \boldsymbol{e} \to \boldsymbol{x}$ という生成モデル
- 教師なし学習では我々は観測値xしか知らないが、背後に確率モデルの潜在変数 z と真の単語ベクトル $\boldsymbol{e}$ がある

$$p(z, \boldsymbol{e}, \boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\phi}) = p(z|\boldsymbol{\theta}) \prod_{i=1}^{\ell} p(\boldsymbol{x}_i|\boldsymbol{e}_i, \boldsymbol{\phi}) p(\boldsymbol{e}_i|z_i, \boldsymbol{\eta})$$

ここで、$p(\boldsymbol{x}_i|\boldsymbol{e}_i)$ は点 $\boldsymbol{x}_i$ だけのディラック測度

$$p(\boldsymbol{x}_i|\boldsymbol{e}_i) = \delta(\boldsymbol{x}_i - \boldsymbol{e}_i)$$

確率モデル (2)



- $z, \boldsymbol{e}_i$ を積分消去すると観測値の尤度が得られる

$$p(\boldsymbol{x}) = \sum_z \left(p(z|\boldsymbol{\theta}) \prod_{i=1}^{\ell} \underbrace{\int p(\boldsymbol{x}_i|\boldsymbol{e}_i) p(\boldsymbol{e}_i|z_i) d\boldsymbol{e}_i}_{p(\boldsymbol{x}_i|z_i)} \right)$$

  - 問題：$\boldsymbol{e}_i$ をそのまま積分するのは困難

変数変換

- 考えを変えて、$\boldsymbol{e}_i \to \boldsymbol{x}_i$ と変数変換する
  - $\boldsymbol{x}_i = f_{\boldsymbol{\phi}}(\boldsymbol{e}_i)$ なので、

$$
\begin{aligned}
p(\boldsymbol{x}_i|z_i, \boldsymbol{\phi}, \boldsymbol{\eta}) &= \int \delta(\boldsymbol{x}_i - f_{\boldsymbol{\phi}}(\boldsymbol{e}_i)) p_{\boldsymbol{\eta}}(\boldsymbol{e}_i|z_i) d\boldsymbol{e}_i \\
&= \int \delta(\boldsymbol{x}_i - \boldsymbol{x}'_i) p_{\boldsymbol{\eta}}(f_{\boldsymbol{\phi}}^{-1}(\boldsymbol{x}'_i)|z_i) \left| \det \frac{\partial f_{\boldsymbol{\phi}}^{-1}}{\partial \boldsymbol{x}'_i} \right| d\boldsymbol{x}'_i \\
&= p_{\boldsymbol{\eta}}(f_{\boldsymbol{\phi}}^{-1}(\boldsymbol{x}_i)|z_i) \left| \det \frac{\partial f_{\boldsymbol{\phi}}^{-1}}{\partial \boldsymbol{x}_i} \right| .
\end{aligned}
$$

- 要するに、単語ベクトルを $f_{\boldsymbol{\phi}}^{-1}$ で逆変換した空間で通常の確率モデルを考えればよい！

変数変換 (例)

- HMMの場合、

$$\log p(\boldsymbol{x}) = \log p_{\mathrm{HMM}}(f_{\phi}^{-1}(\boldsymbol{x})) + \sum_{i=1}^{\ell} \log \left| \det \frac{\partial f_{\phi}^{-1}}{\partial \boldsymbol{x}_i} \right|$$

(論文(7)式)

- ヤコビアン $\log \left| \det \frac{\partial f_{\phi}^{-1}}{\partial \boldsymbol{x}_i} \right|$
  を使って、情報のロスを防いでいる
  …例えばすべてのxに対応するeが原点だと、
  ヤコビアンの値が0になり、log(0)=-∞は無限のコスト

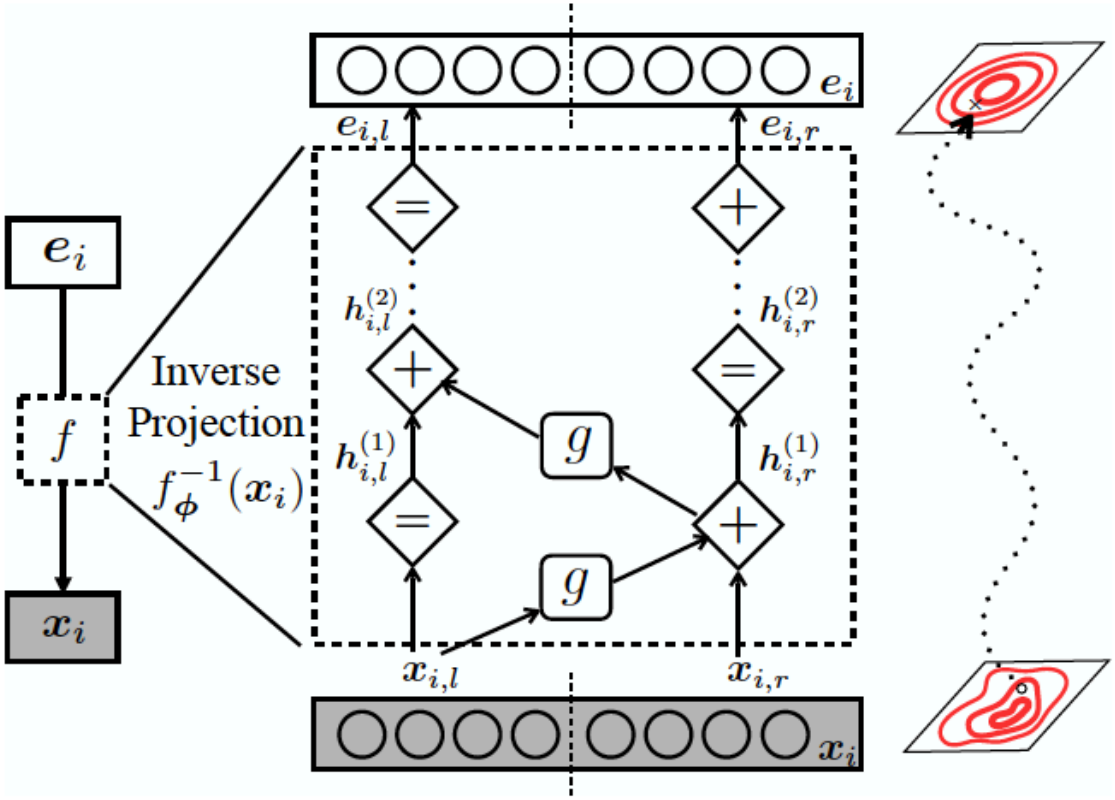
ヤコビアンの計算?

- 変数変換のヤコビアン

$$\left| \det \frac{\partial f_{\boldsymbol{\phi}}^{-1}}{\partial \boldsymbol{x}} \right|$$

  を具体的に計算するのは困難
- ヤコビアンが自動的に1になるような可逆写像のニューラルネット (NICE; Dinh+ 2014)を利用
  - Normalizing flow (Rezende+ 2015他)の考え方と同じ

Volume-preserving invertible projection



(Dinh, Krueger, Bengio 2015)

- レイヤを左右に分け、

$$\mathbf{h}^{(1)}_{i,\ell} = \boldsymbol{x}_{i,\ell}$$

$$\mathbf{h}^{(1)}_{i,r} = \boldsymbol{x}_{i,r} + g(\boldsymbol{x}_{i,\ell})$$

  のような変換を相互に繰り返す
- gはReLUなど
- ヤコビアンの対角要素=1だけが残る→行列式1

Non-linear independent components estimation (NICE)

- 潜在変数の各次元を独立にするような、潜在変数→データへの写像 f を学習したい

$$\mathbf{h} = f(\mathbf{x}); \quad p(\mathbf{h}) = \prod_{d=1}^{D} p(h_d)$$

- ここで変数変換により、

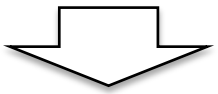
$$p(\mathbf{x}) = p(f(\mathbf{x})) \left| \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right|$$

- 可逆な写像を考えれば、データの生成は

$$\mathbf{h} \sim p(\mathbf{h})$$
$$\mathbf{x} = f^{-1}(\mathbf{h})$$

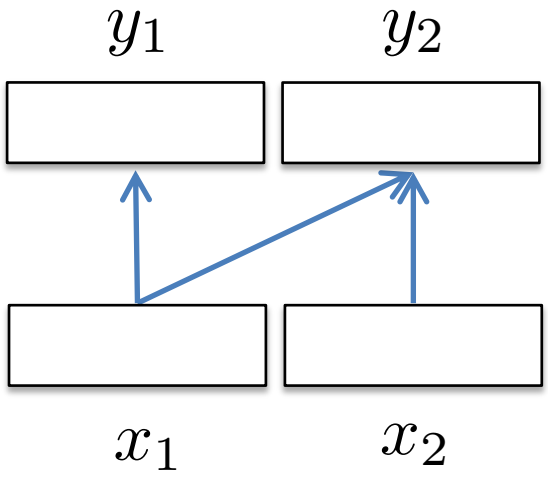
NICE (2)

- ヤコビアン $\left|\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right| = \left|\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right|$ の計算?



三角行列にすることで、自動的に1になるようにする

NICE (3)



- x,y をそれぞれ左右に分けて x=(x1,x2), y=(y1,y2) とし、

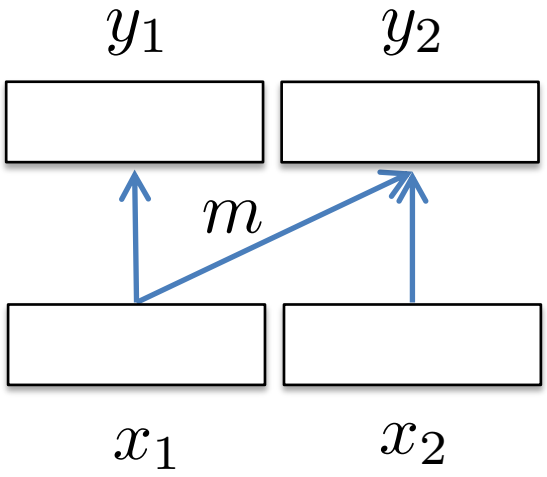
$$y_1 = x_1$$
$$y_2 = x_2 + m(x_1)$$

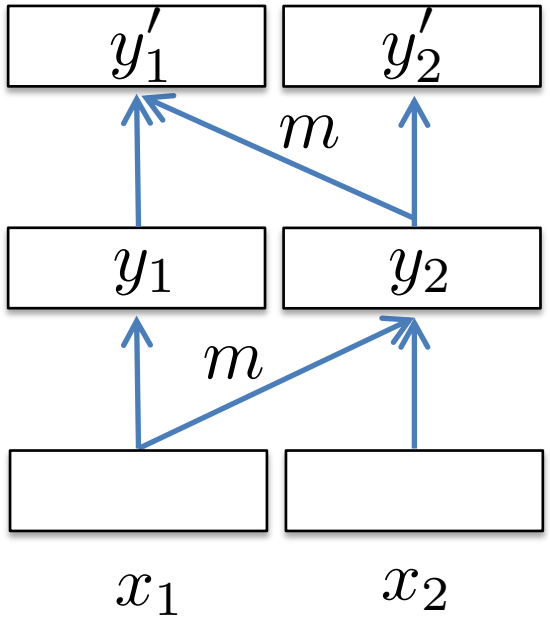
とすれば、

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} I & 0 \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} \end{bmatrix}$$

- よって $\det\left(\frac{\partial y}{\partial x}\right) = 1$!

NICE (4)





- 逆変換も簡単:

$$y_1 = x_1$$
$$y_2 = x_2 + m(x_1)$$

なので、

$$x_1 = y_1$$
$$x_2 = y_2 - m(x_1) = y_2 - m(y_1)$$

- このままだと y_1 が素通しなので、mの方向を逆にして層を重ねる
  - 経験的に3層以上が必要、論文の実験では4層

実験

- 代表的な教師なし学習である教師なし品詞学習と教師なし構文解析で評価
  - 教師なし品詞学習：HMM (過去研究多数)
  - 教師なし構文解析：DMV (Klein and Manning 2004)

  が確率モデル
- もちろん、他のタスクでも同様に使える
- 上記のタスクでも、最近の進んだモデルを使うことはできるが、そことは直交するので基本モデルで評価

教師なし品詞学習

- WSJの標準splitで評価
- 窓幅1のword2vecで単語ベクトルを事前学習
  - 品詞学習では意味よりも文法が重要なため
  - 単語ベクトルの次元＝100
  - WSJに加えて、1 billion words dataset (Chelba+13)を使って学習
- Projectorのcoupling layerは4, 8, 16で実験
  - 基本設定はNICEの論文と同じ

教師なし品詞学習 (2)

System	M-1	VM
w/o hand-engineered features		
Discrete HMM	62.7	53.8
PYP-HMM (Blunsom and Cohn, 2011)	77.5	69.8
NHMM (basic) (Tran et al., 2016)	59.8	54.2
NHMM (+ Conv) (Tran et al., 2016)	74.1	66.1
NHMM (+ Conv & LSTM) (Tran et al., 2016)	79.1	71.7
Gaussian HMM (Lin et al., 2015)	75.4 (1.0)	68.5 (0.5)
Ours (4 layers)	79.5 (0.9)	73.0 (0.7)
Ours (8 layers)	**80.8** (1.3)	**74.1** (0.7)
Ours (16 layers)	73.2 (4.3)	70.5 (2.1)

- Gaussian HMMを抜いて最高性能を達成
  - NHMMの最後はLSTMを併用しているのに注意
  - Projectorは複雑すぎない方が、性能が高い

教師なし品詞学習 (3)

Ours (4 layers)	79.5 (0.9)	73.0 (0.7)
Ours (8 layers)	**80.8** (1.3)	**74.1** (0.7)
Ours (16 layers)	73.2 (4.3)	70.5 (2.1)
w/ hand-engineered features		
Feature HMM (Berg-Kirkpatrick et al., 2010)	75.5	–
Brown (+ proto) (Christodoulopoulos et al., 2010)	76.1	68.8
Cluster (word-based) (Yatbaz et al., 2012)	80.2	72.1
Cluster (token-based) (Yatbaz et al., 2014)	79.5	69.1

- 素性を手で設計したモデルと比べても、最高性能

教師なし構文解析

- DMVの各文法カテゴリに、(真の単語ベクトルを生成する)多変量ガウス分布が存在
- 通常の教師なし構文解析は、単語から始めない
  - 正解の品詞から始める
  - 単語を直接使うと、モデルの次元が多すぎるため
- 提案法では、単語ベクトルの情報が直接使える
  → 単語から構文解析が始められる
  - 単語情報を使って、より精緻な構文解析が期待できる (cf. with a hat / with a telescope)
  - 比較のため、ベースラインは前で求めた教師なし品詞から始めている

教師なし構文解析 (2)

- 文長10以下と、全ての場合に分けて評価
- ここでも最高精度を達成

System	⩽ 10	all
w/o gold POS tags		
DMV (Klein and Manning, 2004)	49.6	35.8
E-DMV (Headden III et al., 2009)	52.1	38.2
UR-A E-DMV (Tu and Honavar, 2012)	58.9	46.1
CS* (Spitkovsky et al., 2013)	72.0*	64.4*
Neural E-DMV (Jiang et al., 2016)	55.3	42.7
CRFAE (Cai et al., 2017)	37.2	29.5
Gaussian DMV	55.4 (1.3)	43.1 (1.2)
Ours (4 layers)	58.4 (1.9)	46.2 (2.3)
Ours (8 layers)	**60.2** (1.3)	**47.9** (1.2)
Ours (16 layers)	54.1 (8.5)	43.9 (5.7)

これは句読点情報を使っているのでチート

教師なし構文解析 (3)

- 正解の品詞から始めると、精度は相当に高い
  - 正解の品詞に、正しい構文解析の情報が含まれているため
- モデルにも改善は色々ある

Ours (4 layers)	58.4 (1.9)	46.2 (2.3)
Ours (8 layers)	**60.2** (1.3)	**47.9** (1.2)
Ours (16 layers)	54.1 (8.5)	43.9 (5.7)
w/ gold POS tags (for reference only)		
DMV (Klein and Manning, 2004)	55.1	39.7
UR-A E-DMV (Tu and Honavar, 2012)	71.4	57.0
MaxEnc (Le and Zuidema, 2015)	73.2	65.8
Neural E-DMV (Jiang et al., 2016)	72.5	57.6
CRFAE (Cai et al., 2017)	71.7	55.7
L-NDMV (Big training data) (Han et al., 2017)	77.2	63.2

Caveats

- 初期化と単語ベクトルの選択で性能は若干変化する

System	M-1	VM
Ours (4 layers)	78.2	71.2
Ours (8 layers)	72.5	69.7
Ours (16 layers)	67.2	69.2

Table 3: Unsupervised POS tagging results of our approach on WSJ, with random initialization of syntax model.

System	$\leqslant 10$	all
Gaussian DMV	53.6	41.3
Ours (4 layers)	56.9	43.9
Ours (8 layers)	57.1	42.3
Ours (16 layers)	52.9	39.5

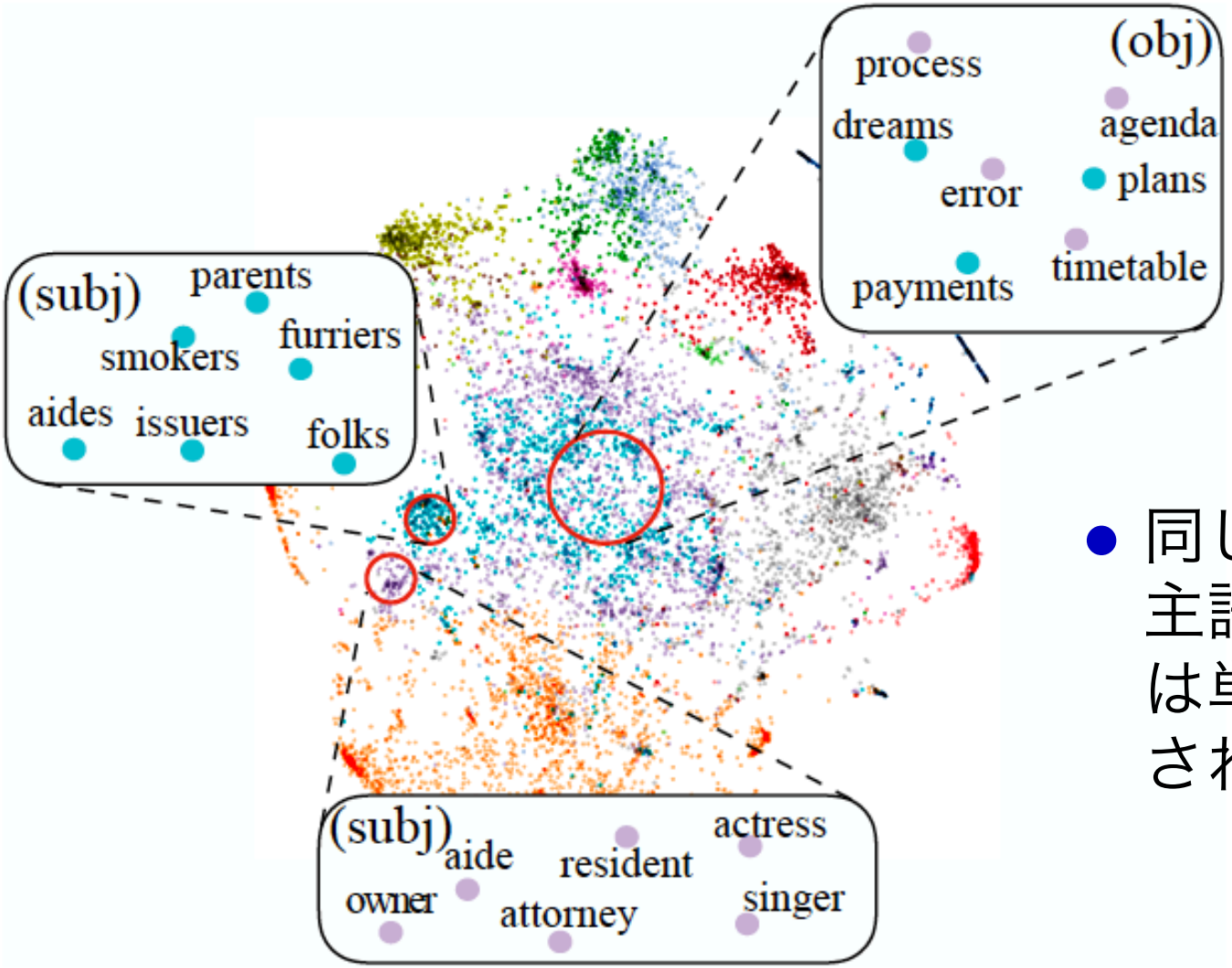
Table 5: Directed dependency accuracy on section 23 of WSJ, with fastText vectors as the observed embeddings.

HMMで学習されたxとe

Target	Skip-gram	Markov Structure
come	go came follow coming sit	be go do give follow
singing	dancing sing drumming dance dances	dancing drumming marching playing recording
cigars	cigarettes sodas champagne cigar rum	sodas bottles drinks pills cigarettes
newer	flashier fancier conventional low-end new-generation	softer lighter thinner darker smoother
fanciest	priciest up-scale loveliest fancier high-end	liveliest priciest smartest best-run fastest-growing

- xは窓幅1で学習されているが、真に文法的ではない

DMVで学習された“真の単語ベクトル”



- 同じ名詞でも、主語になる名詞は単複が区別されている

参考実装

- https://github.com/jxhe/struct-learning-with-flow

まとめ

- 通常の確率モデルと単語ベクトルを繋ぐ、巧妙な方法
- 可逆写像を利用することで、変換後の単語ベクトルを直接使って確率モデルを学習できる
- ニューラルネットの表現力と、確率モデルの構造化を両方備えたモデル化が可能
  - 実際のタスクでも、適当な方法に比べて最高精度