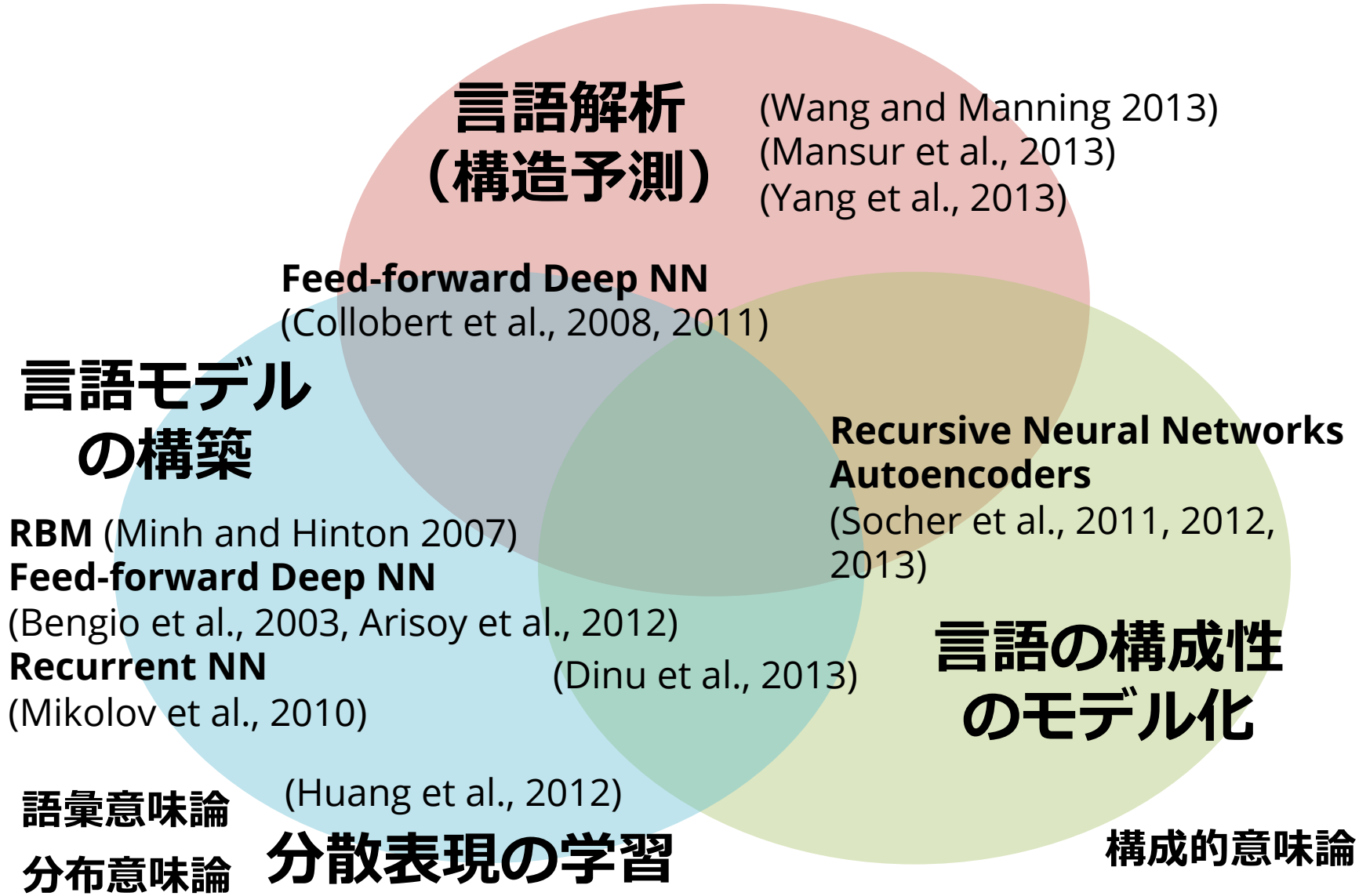
自然言語処理分野におけるディープラーニングの現状

渡邉　陽太郎
東北大学大学院情報科学研究科

IBIS2013 企画セッション2：ディープラーニング
2013/11/12

NLPにおけるディープラーニング



自然言語処理分野でディープラーニングを導入するモチベーション

- **分散表現（Distributed Representation）**
  - 単語やフレーズを固定長のベクトルで表現
  - 単語やフレーズの持つ言語的な性質を含み、類似する単語が類似したベクトルを持つよう学習
- **言語の構成性のモデル化**
  - 句や文の意味を、それを構成する単語の合成により得る
- **表現学習 (Representation Learning)**
  - 素性エンジニアリングを最小限に
- **マルチタスク学習**
  - あるタスクから得られるパラメータを別のタスクに利用

目次

- 言語処理分野におけるディープラーニング
- Deep NNに基づく構造予測
  - 言語解析のための統一的フレームワーク
- 言語モデル・単語の分散表現
- Deep NNを用いた言語の構成性のモデル化
- 言語処理でディープアーキテクチャは必要か？
- まとめ

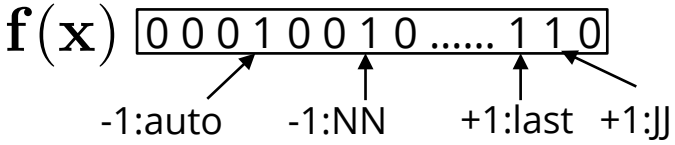
NLPの構造予測問題

	The	luxury	auto	maker	last	year	sold	1,214	cars	in	the	U.S.
品詞タグ付け	DT	NN	NN	NN	JJ	NN	VBD	CD	NNS	IN	DT	NNP
基本句チャンキング	B-NP	I-NP	I-NP	E-NP	B-NP	E-NP	B-VP	B-NP	E-NP	S-PP	B-NP	E-NP
固有表現抽出	B-ORG	I-ORG	I-ORG	E-ORG	O	O	O	O	O	O	B-LOC	E-LOC
意味役割付与	-	B-A1	E-A1	S-A0 **PRED**	-	-	-	-	-	-	-	-
	B-A0	I-A0	I-A0	I-A0	B-AM-TMP	- **PRED**		B-A1	I-A1	B-AM-LOC	...	

- **単語列に対して、適切な系列を出力する**
  - 条件付確率場 (Lafferty+ 2011), 構造化パーセプトロン等
  - 各タスクごとに素性を人手で設計し、モデルを構築することが一般的

Deep NNでの単語の表現方法: 分散表現 (Distributed Representation)

- **従来の表現方法**



タスクによっては数百万、数千万次元

- **表現の汎化**

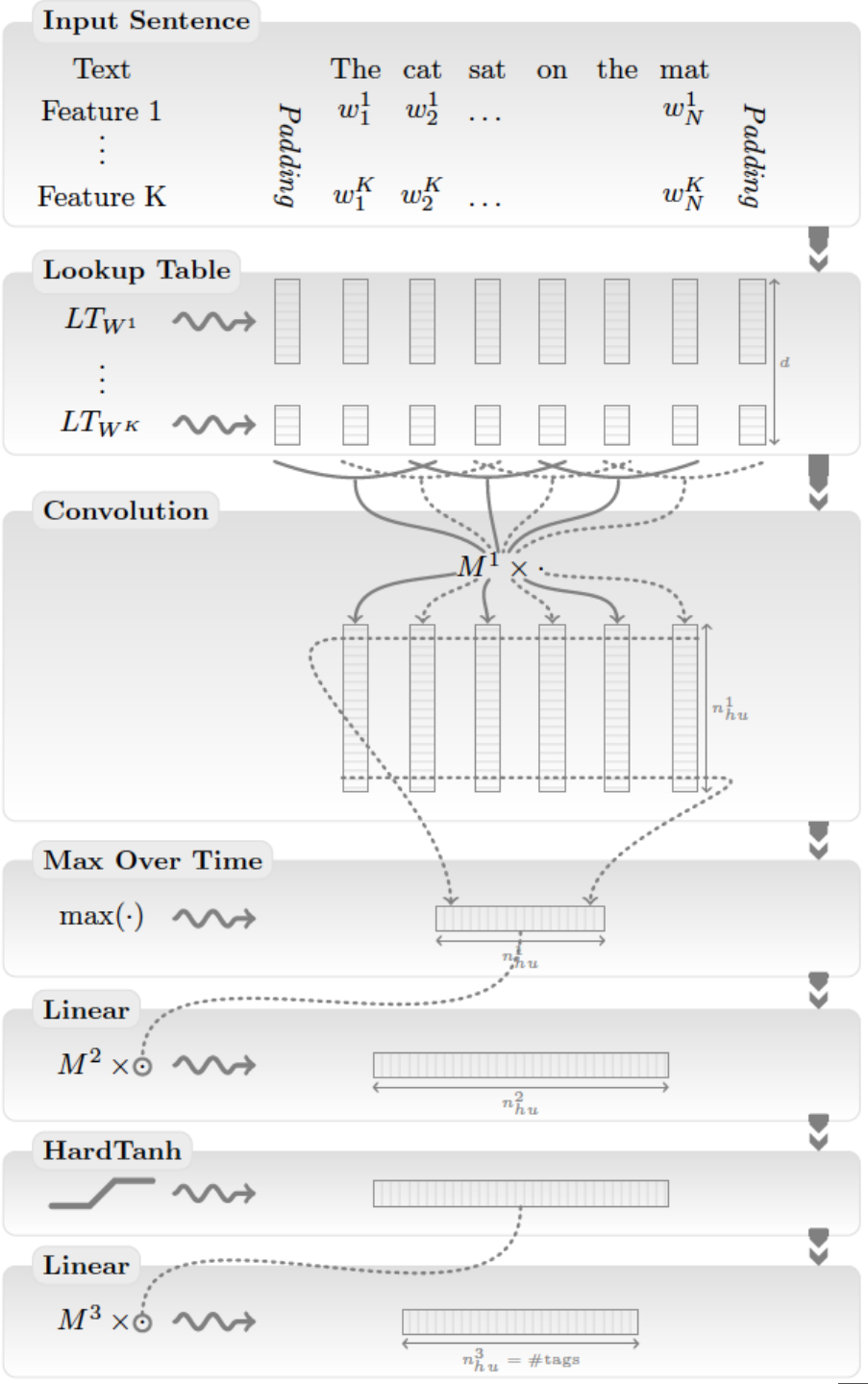
The cat is walking in the bedroom ...

A dog was running in the bedroom ...

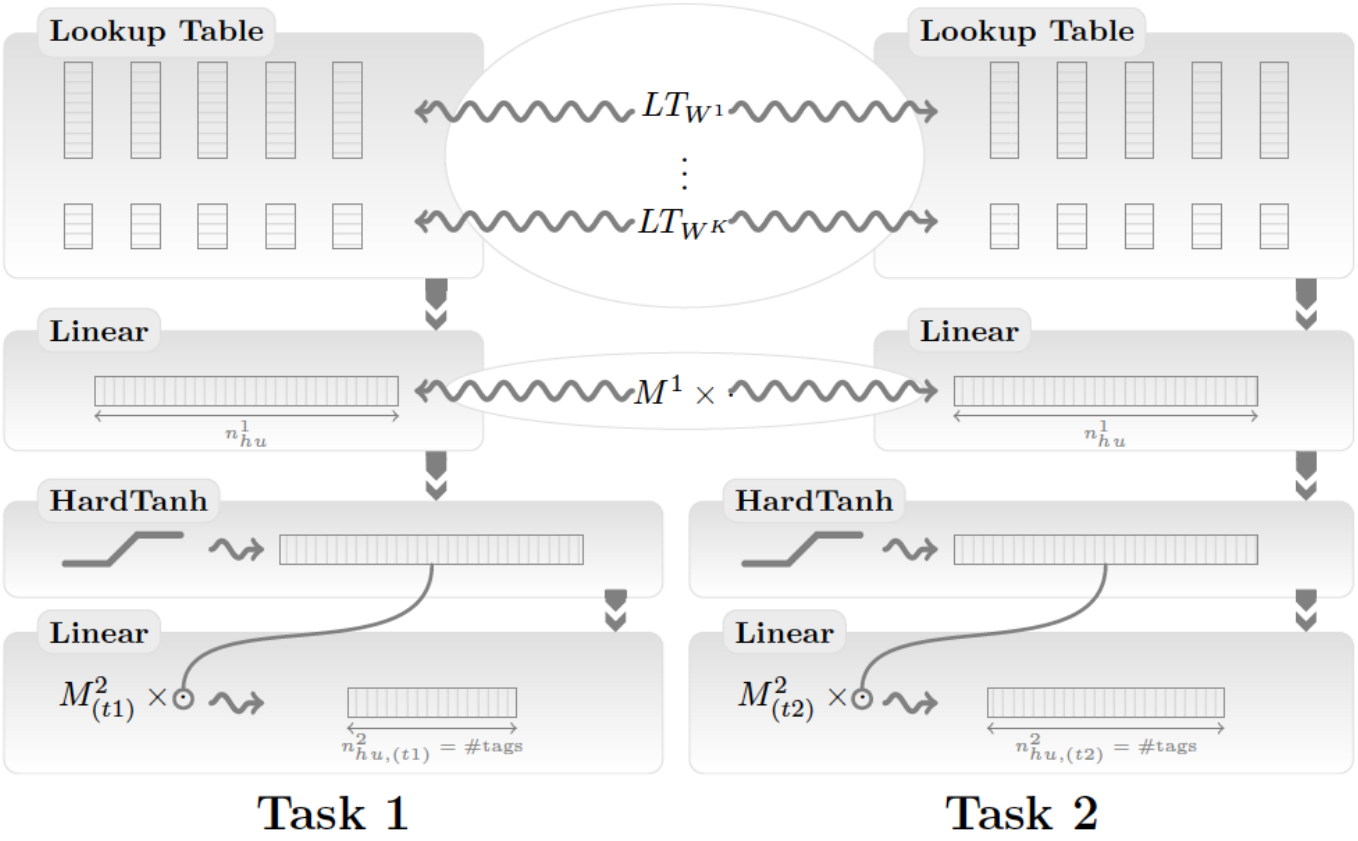
  - 同一のカテゴリの単語 (cat-dog, walking-running, the-a) を類似した単語とみなしたい
- **固定長の低次元ベクトルとして単語を表現**
  - 次元数：e.g. 50, 100, 300, 500, ...
  - 単語が持っている意味のある側面が共通していれば、ベクトルの要素の一部が類似して欲しい

A Unified Architecture for NLP
(Collobert et al., 2008, 2011)

- **NLPタスクを統一的に扱う多層NN**
  - 品詞タグ付け、基本句のチャンキング、固有表現抽出、意味役割付与、言語モデル
- **限定された特徴量**
  - 単語、先頭が大文字かどうかなど
- **表現学習**
  - 畳み込みで周辺文脈を考慮、プーリングで必要な情報を選択



Deep NNを用いたマルチタスク学習

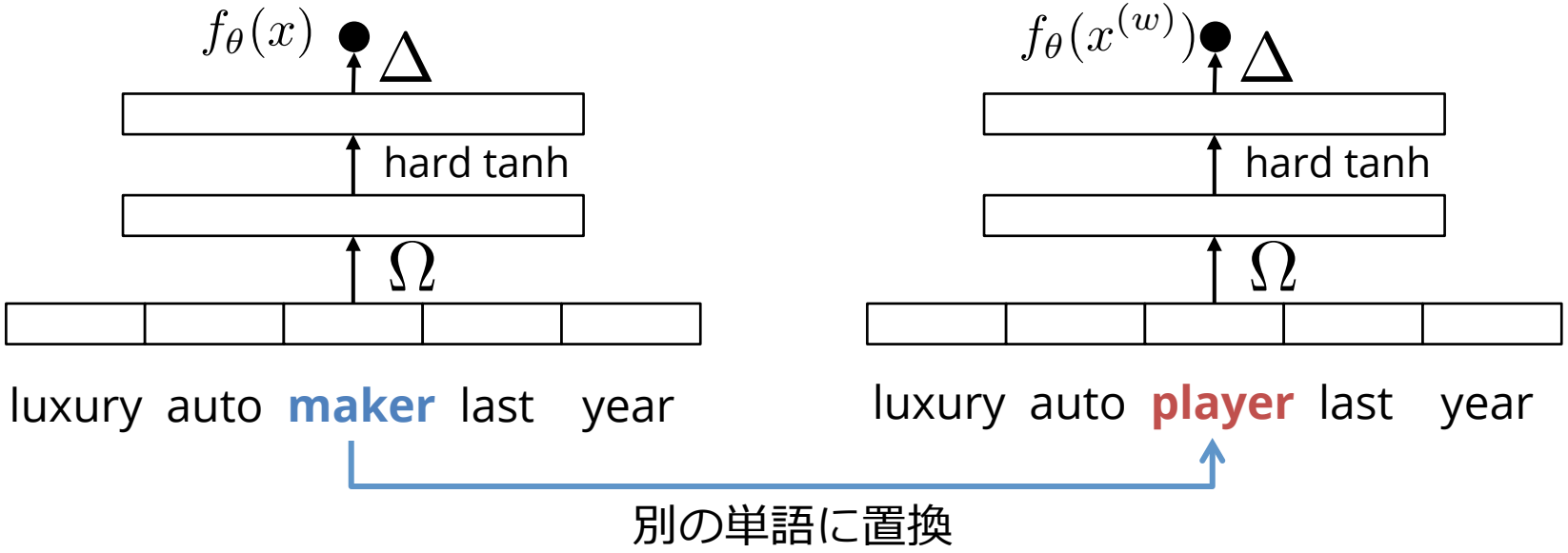


- 単語の分散表現、その下の隠れ層のパラメータを複数タスク間で共有
  - 一方のタスクで得られた情報を別のタスクに用いる

言語モデルの学習 (Collobert et al., 2008)

… luxury auto ___?___ last year …

- **Negative exampleをサンプリング（擬似負例）**
  - Contrastive Estimation (Smith and Eisner 2005)
- **ランキングに基づく学習**
  - 損失関数: $\sum_{x \in \mathcal{X}} \sum_{w \in \mathcal{D}} \max\{0, 1 - f_\theta(x) + f_\theta(w^{(x)})\}$



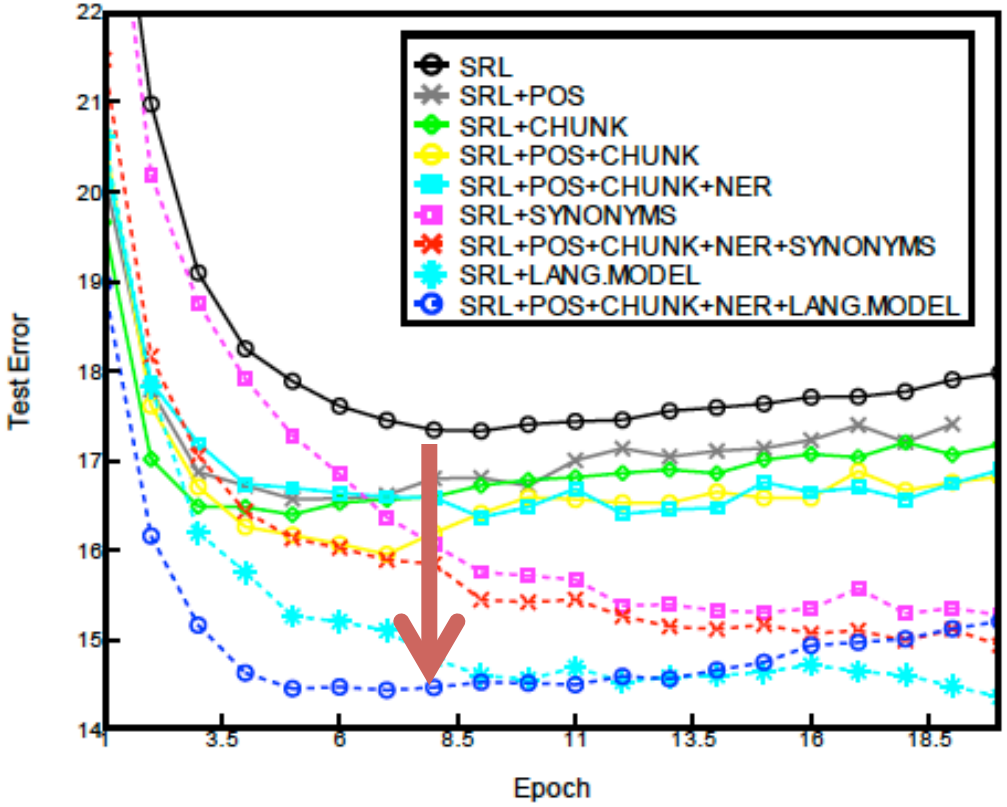
言語モデル学習結果

各クエリ単語についてユークリッド距離の高い上位10件の単語を見ると…

国	神	ゲーム機	色	イベント（打つ,壊す,削る…）	単位
FRANCE	JESUS	XBOX	REDDISH	SCRATCHED	MEGABITS
454	1973	6909	11724	29869	87025
AUSTRIA	GOD	AMIGA	GREENISH	NAILED	OCTETS
BELGIUM	SATI	PLAYSTATION	BLUISH	SMASHED	MB/S
GERMANY	CHRIST	MSX	PINKISH	PUNCHED	BIT/S
ITALY	SATAN	IPOD	PURPLISH	POPPED	BAUD
GREECE	KALI	SEGA	BROWNISH	CRIMPED	CARATS
SWEDEN	INDRA	PSNUMBER	GREYISH	SCRAPED	KBIT/S
NORWAY	VISHNU	HD	GRAYISH	SCREWED	MEGAHERTZ
EUROPE	ANANDA	DREAMCAST	WHITISH	SECTIONED	MEGAPIXELS
HUNGARY	PARVATI	GEFORCE	SILVERY	SLASHED	GBIT/S
SWITZERLAND	GRACE	CAPCOM	YELLOWISH	RIPPED	AMPERES

類似した意味を持つ単語が近い場所に集まっている

マルチタスク学習結果
(Collobert and Weston 2008)

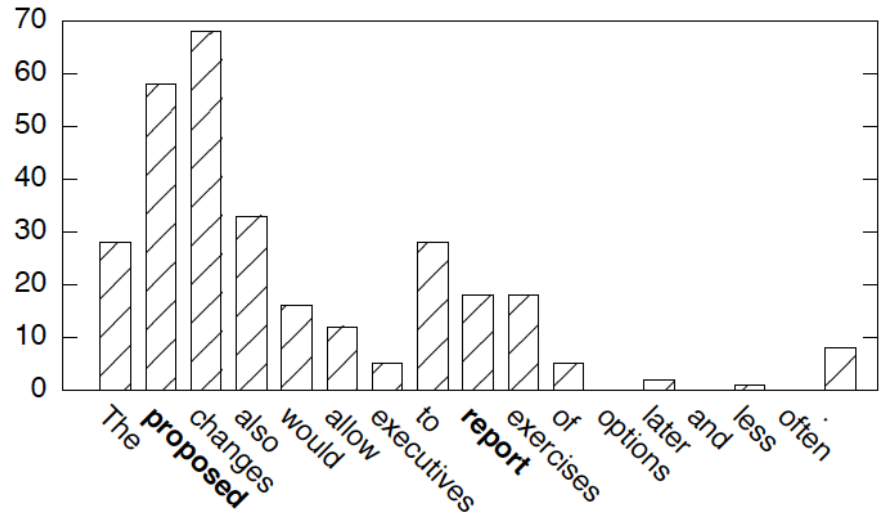


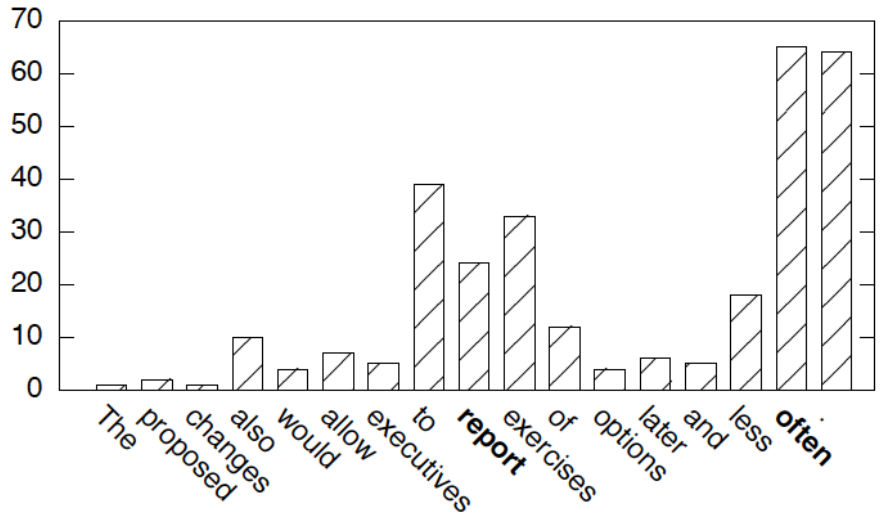
Task		Benchmark	SENNA
Part of Speech (POS)	(Accuracy)	97.24 %	97.29 %
Chunking (CHUNK)	(F1)	94.29 %	94.32 %
Named Entity Recognition (NER)	(F1)	89.31 %	89.59 %
Parse Tree level 0 (PT0)	(F1)	91.94 %	92.25 %
Semantic Role Labeling (SRL)	(F1)	77.92 %	75.49 %

- **意味役割付与タスクで評価**
- **マルチタスク学習でエラーレート低下**
  - 言語モデルとの組み合わせで最も良い性能
- **限られた特徴量でstate-of-the-artな性能を達成**
  - 表現学習の結果

SENNA: http://ronan.collobert.com/senna/

畳み込み+Max Poolingによる効果





- **意味役割付与タスクでの振る舞い**
  - Max Pooling Layerで選択された値に対応する単語を頻度をカウント
  - 注目している述語が異なると、選ばれる値が異なる

Deep NNを利用するメリット

- **素性抽出の時間を削減可能**
  - 従来手法では、特徴ベクトルの作成には、特徴として利用する文字列を抽出、IDに変換する必要がある
  - Deep NNは単語情報（＋α）を見て対応するベクトルをlookupするだけ

システムの使用メモリと解析スピード

POS System	RAM (MB)	Time (s)
Toutanova et al. (2003)	800	64
Shen et al. (2007)	2200	833
SENNA	32	4

SRL System	RAM (MB)	Time (s)
Koomen et al. (2005)	3400	6253
SENNA	124	51

目次

- 言語処理分野におけるディープラーニング
- Deep NNに基づく構造予測
- 言語モデル・単語の分散表現
  - Feed-forward Deep NN, Recurrent NN
  - 言語モデルの比較、学習語の分散表現に関する調査
- Deep NNを用いた言語の構成性のモデル化
- 言語処理でディープアーキテクチャは必要か？
- まとめ

言語モデル

The luxury auto maker last year ___?___

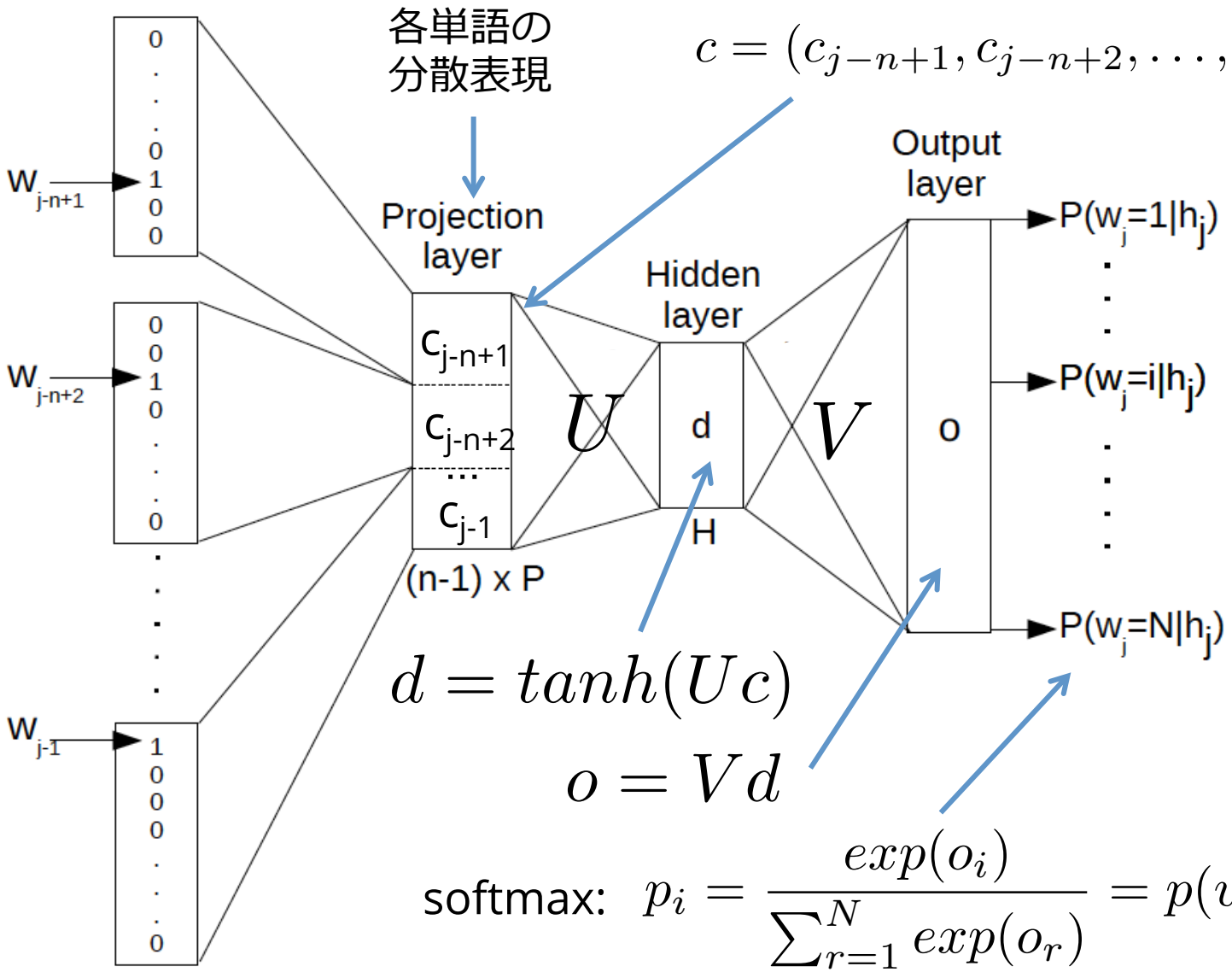
- **過去の履歴から次の単語を予測 (word prediction)**

$$\hat{P}(w_1^T) = \prod_{t=1}^{T} \hat{P}(w_t|w_1^{t-1})$$

$$\hat{P}(w_t|w_1^{t-1}) \approx \hat{P}(w_t|w_{t-n+1}^{t-1})$$

  - Bigram model: P (sold | year)
  - Trigram model: p (sold | last year)

Feed-forward NN Language Model (Bengio et al., 2003, Schwenk 2007, Arisoy et al., 2012)



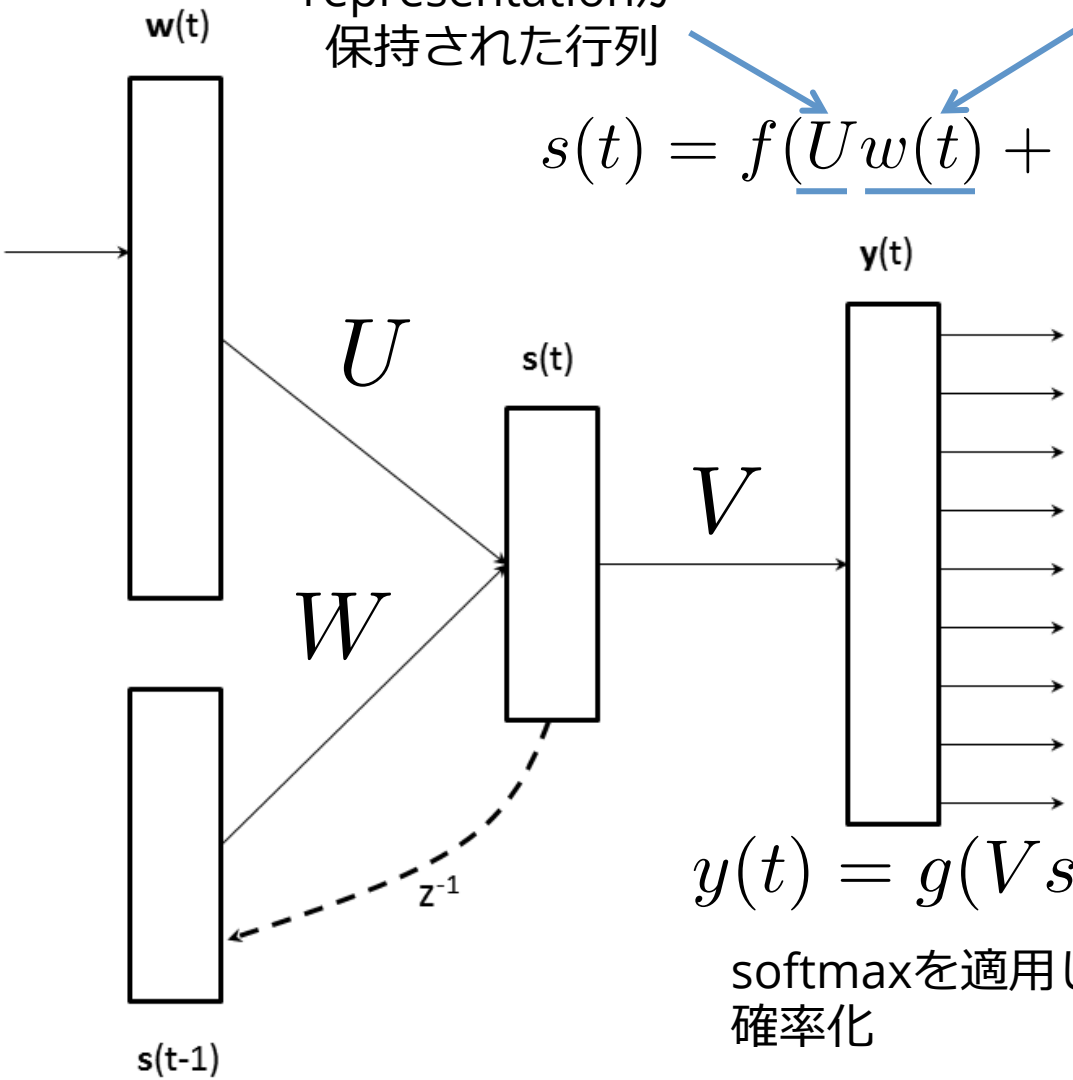
Recurrent Neural Network Language Model (Mikolov et al., 2010)

各単語のword representationが保持された行列

注目している単語だけ1になっているベクトル

$$s(t) = f(Uw(t) + Ws(t-1))$$

過去の履歴（文脈）を保持しているベクトル（直前のsの値）



$$y(t) = g(Vs(t))$$

softmaxを適用して確率化

Feed-forward NN-based LMよりも広いコンテキストを見ることができる

Feed-forward (Deep) Neural Networks vs. Recurrent Neural Networks (Arisoy et al., 2012)

隠れ層の次元　　分散表現の次元

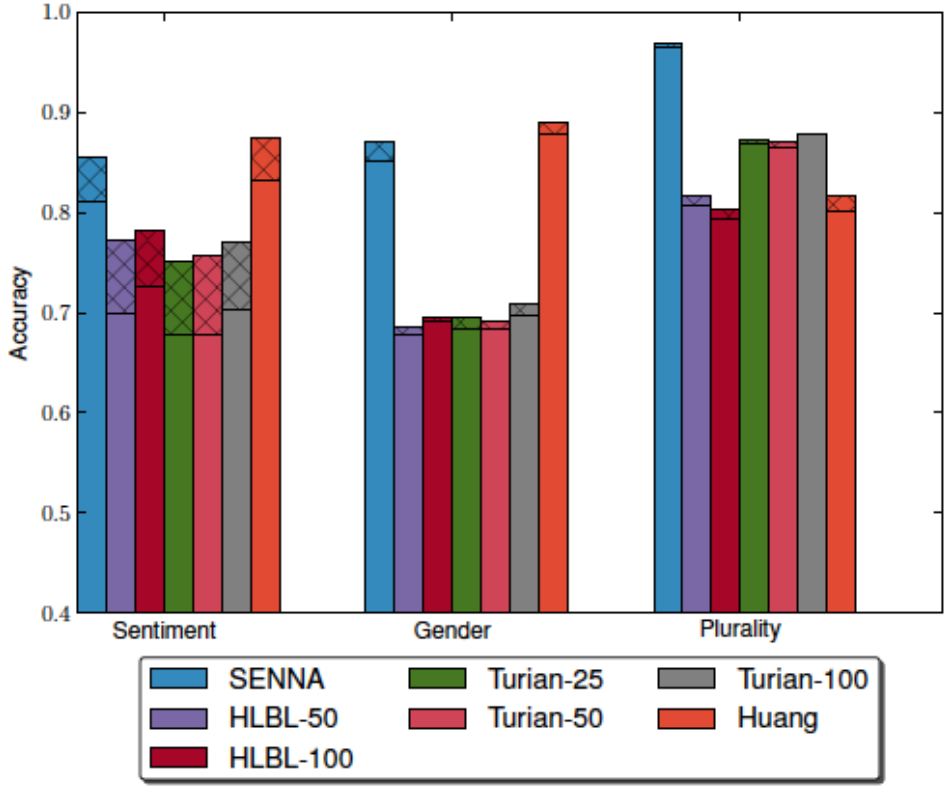
Models	Perplexity	WER(%)
4-gram LM	114.4	22.3
DNN LM: h=500, d=30		
with 1 layer (NNLM)	115.8	22.0
with 4 layers	108.0	21.6
DNN LM: h=500, d=60		
with 1 layer (NNLM)	109.3	21.5
with 3 layers	105.0	21.3
DNN LM: h=500, d=120		
with 1 layer (NNLM)	104.0	21.2
with 3 layers	102.8	20.8
Model M (Chen, 2008)	99.1	20.8
RNN LM (h=200)	99.8	-
RNN LM (h=500)	83.5	-

- Feed-forward NNは隠れ層を増やすことで性能向上
- 分散表現を大きくするとパープレキシティ低下
- Recurrent NNの方がパフォーマンスが良い

学習された分散表現に、どの程度の情報が含まれているのか？

	Sentiment		Noun Gender		Plurality	
	Positive	Negative	Feminine	Masculine	Plural	Singular
Samples	good	bad	Ada	Steve	cats	cat
	talent	stupid	Irena	Roland	tables	table
	amazing	flaw	Linda	Leonardo	systems	system

	Synonyms and Antonyms		Regional Spellings	
	Synonyms	Antonyms	UK	US
Samples	store shop	rear front	colour	color
	virgin pure	polite impolite	driveable	drivable
	permit license	friend foe	smash-up	smashup

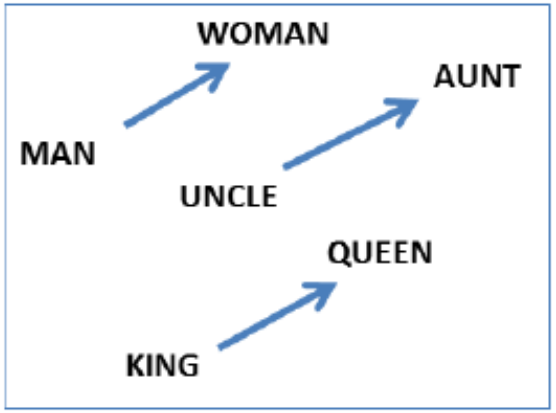


- 学習された分散表現の表現能力調査　(Chen et al. 2013)
- 分散表現を入力としてSVMで分類
- 高いものでは90%近く、またはそれを上回る性能を達成
- 学習された分散表現に単語の持つ性質が含まれている

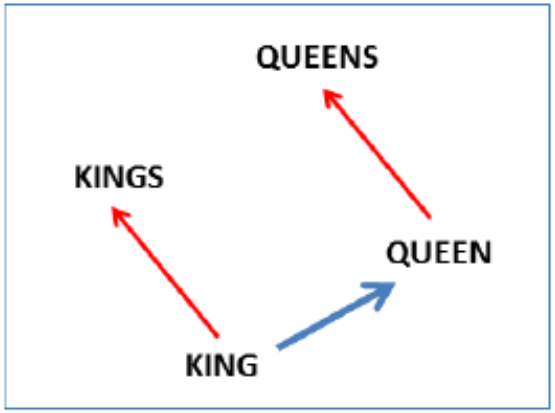
分散表現を用いて類推の問題を解く
(Mikolov et al. 2013)

構文に関する類推問題
- 形容詞
  <good – better> <rough - ?>
- 名詞
  <year – years> <law - ?>
- 動詞
  <see – sees> <return - ?>

Method	Adjectives	Nouns	Verbs	All
LSA-80	9.2	11.1	17.4	12.8
LSA-320	11.3	18.1	20.7	16.5
LSA-640	9.6	10.1	13.8	11.3
RNN-80	9.3	5.2	30.4	16.2
RNN-320	18.2	19.0	45.0	28.5
RNN-640	21.0	25.2	54.8	34.7
RNN-1600	**23.9**	**29.2**	**62.2**	**39.6**



性別の関係

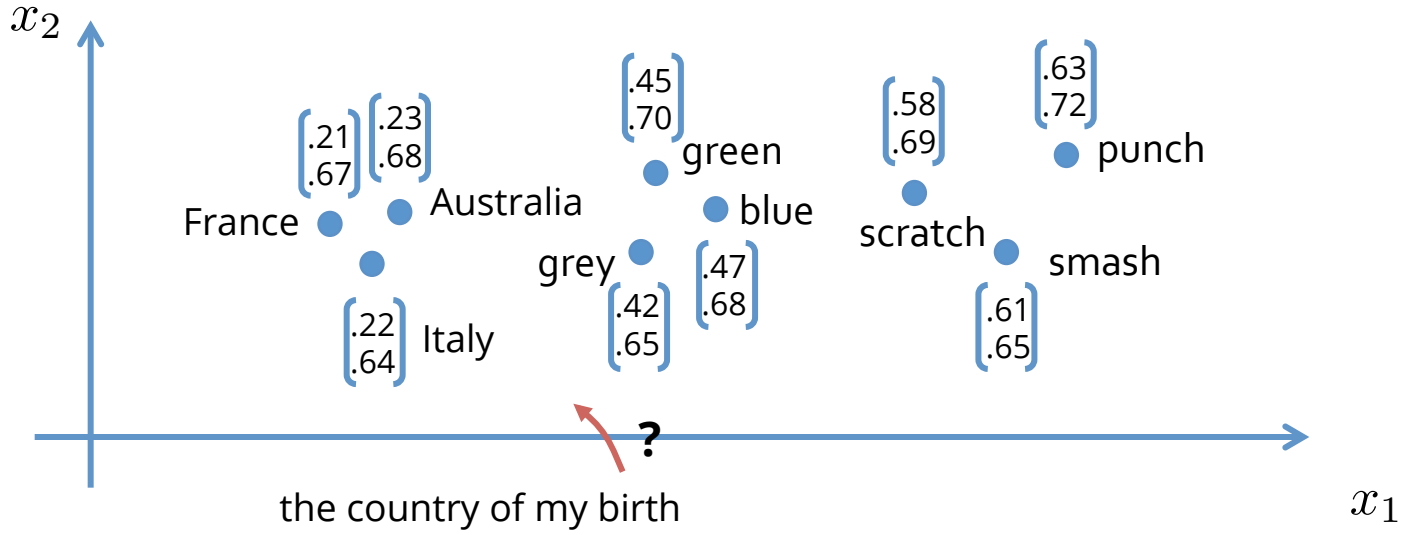


kings – king + queen ≒ queens

目次

- 言語処理分野におけるディープラーニング
- Deep NNに基づく構造予測
- 言語モデル・単語の分散表現
- Deep NNを用いた言語の構成性のモデル化
  - Recursive Neural Networksとその拡張
  - 言語解析への応用例
- 言語処理でディープアーキテクチャは必要か？
- まとめ

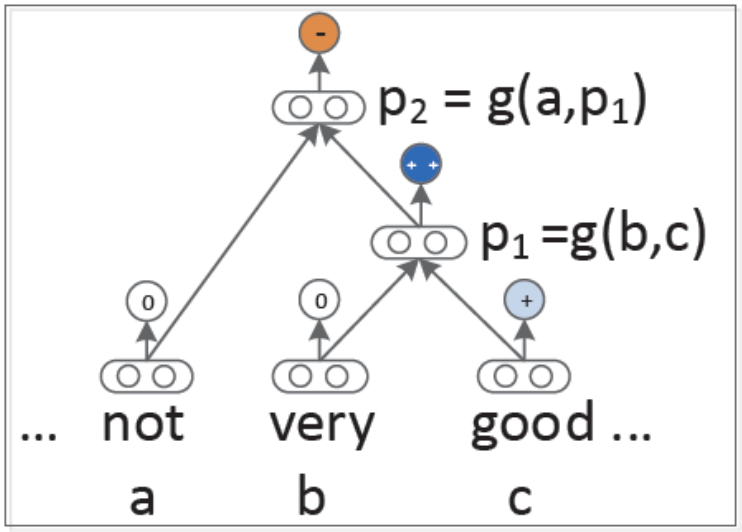
単語はベクトル表現を得られる。ではフレーズは？



- **構成性原理（principle of compositionality）**
  - 句や文の意味（ベクトル）は、それを構成する単語と合成手続きにより決定される
- **句や文の分散表現はどう得たら良いのか？**
  - 何らかの合成関数を用いて同一のベクトル空間にマップする

Recursive Neural Networks

(Goller and Küchler 1996; Socher+ 2011)



$$p_1 = f\left(W\begin{bmatrix} b \\ c \end{bmatrix}\right)$$

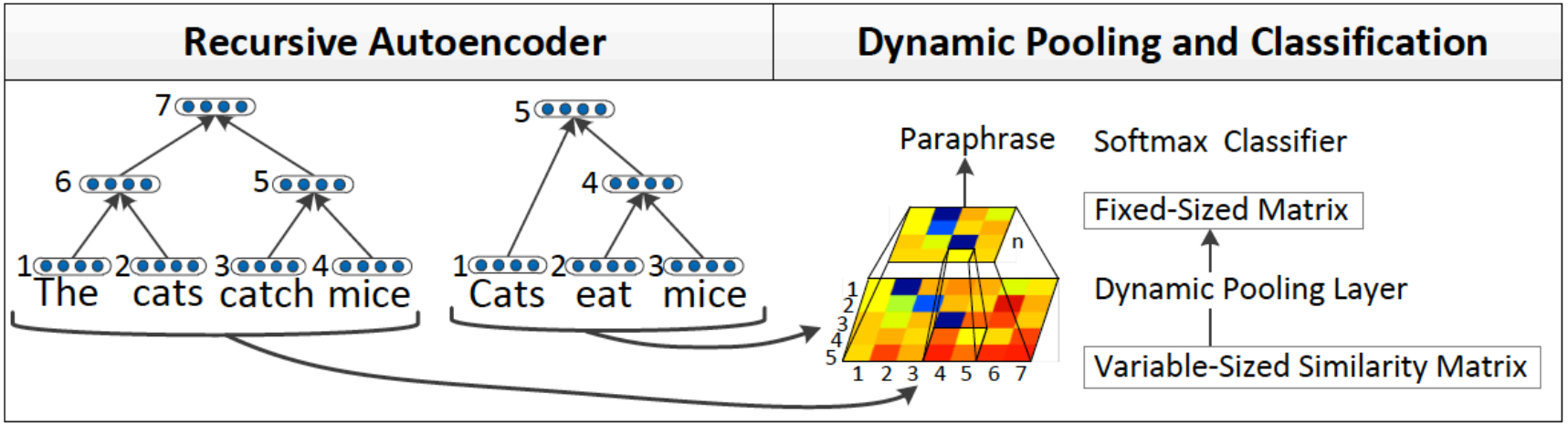
$$p_2 = f\left(W\begin{bmatrix} a \\ p_1 \end{bmatrix}\right)$$

$$W \in \mathbb{R}^{d\times 2d} \quad f = \tanh$$

- **句や文の意味表現を構成的に得る一手法として提案**
  - 2つのベクトルを組み合わせて1つの新たなベクトルを得る
  - 同一の次元数を持つので、単語と句、文が比較可能に
- **合成方法**
  - 子ベクトルをパラメータ行列Wと活性化関数を用いて合成
  - 合成後のベクトルも、合成前のベクトルと同じ次元数 ⇒ 再帰的な適用が可能

RNN (Recursive AE) 応用: 換言認識

(Socher+ 2011)

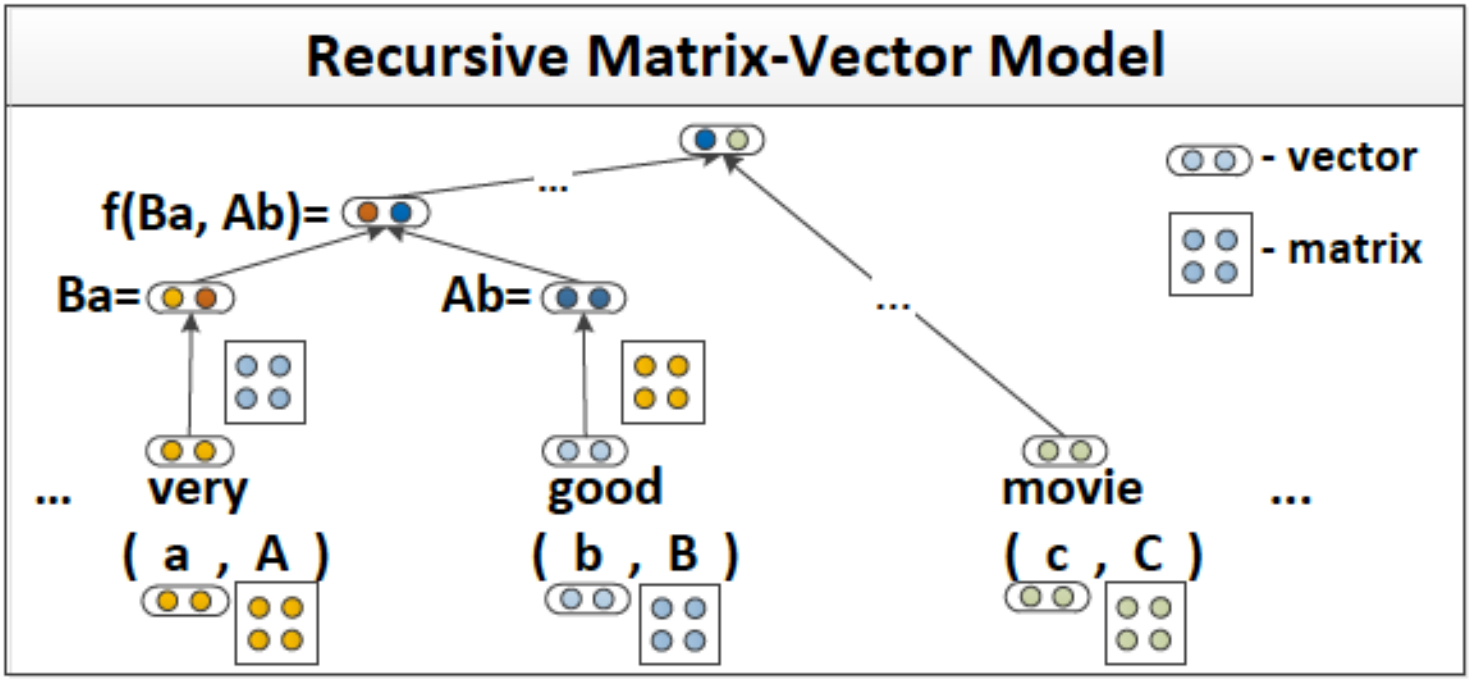


- Autoencoderの学習: 新聞記事（数万分程度）を利用
- 評価: MSR 言い換えコーパス
- 限られた特徴量でstate-of-the-artな性能を達成

Model	Acc.	F1
All Paraphrase Baseline	66.5	79.9
Rus et al. (2008) [16]	70.6	80.5
Mihalcea et al. (2006) [17]	70.3	81.3
Islam and Inkpen (2007) [18]	72.6	81.3
Qiu et al. (2006) [19]	72.0	81.6
Fernando and Stevenson (2008) [20]	74.1	82.4
Wan et al. (2006) [21]	75.6	83.0
Das and Smith (2009) [15]	73.9	82.3
Das and Smith (2009) + 18 Features	76.1	82.7
Unfolding RAE + Dynamic Pooling	**76.8**	**83.6**

MV-RNN: Matrix-Vector RNN

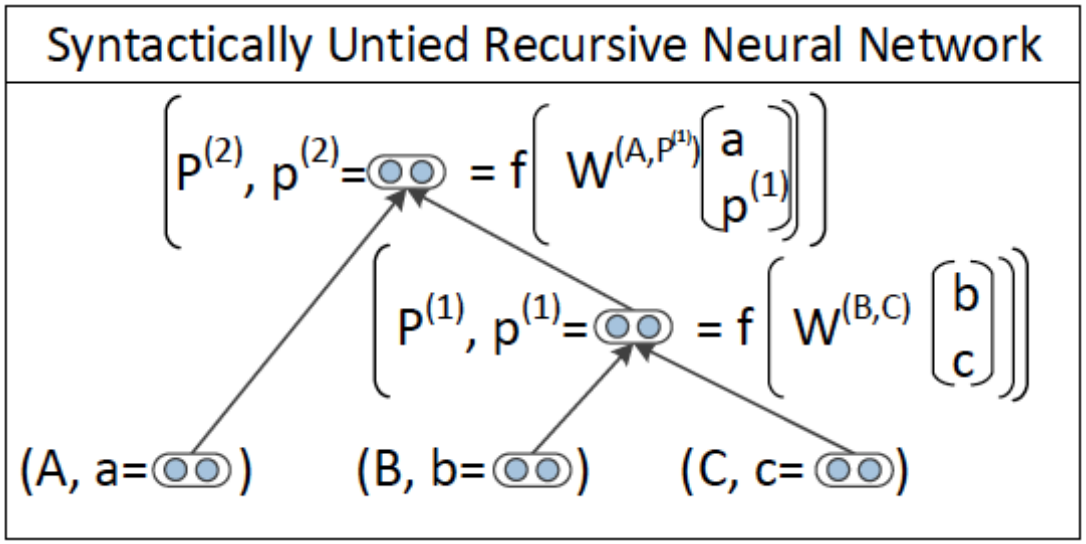
(Socher et al., 2012)



$$p = f_{A,B}(a,b) = f(Ba, Ab) = g\left(W \begin{bmatrix} Ba \\ Ab \end{bmatrix}\right)$$

- 単語や句の表現ごとに振るまいを変えられるようにRNNを拡張 ⇒ ベクトルと行列を用いて単語や句を表現
- **ベクトル**：どういう意味を持つか、**行列**：意味をどう変更するか

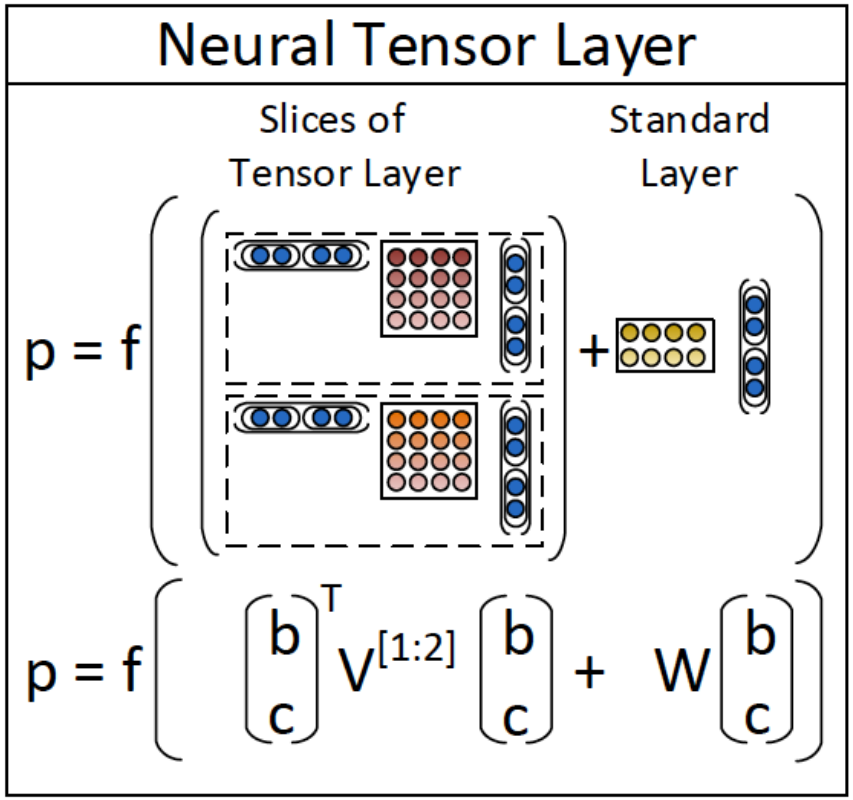
SU-RNN: Syntactically Untied RNN (Socher et al., 2013)



Parser	dev (all)	test≤ 40	test (all)
Stanford PCFG	85.8	86.2	85.5
Stanford Factored	87.4	87.2	86.6
Factored PCFGs	89.7	90.1	89.4
Collins			87.7
SSN (Henderson)			89.4
Berkeley Parser			90.1
CVG (RNN)	85.7	85.1	85.0
CVG (SU-RNN)	91.2	91.1	90.4
Charniak-SelfTrain			91.0
Charniak-RS			92.1

- **確率的文脈自由文法(PCFG)とモデルと識別モデルの組合せ**
- **構文カテゴリを考慮したパラメータの分割**
  - 構文カテゴリ：冠詞, 形容詞, 名詞, 動詞, etc.
  - カテゴリの違いによる振る舞いを捉える
- **構文カテゴリを考慮しないRNNと比較して大幅な性能改善**

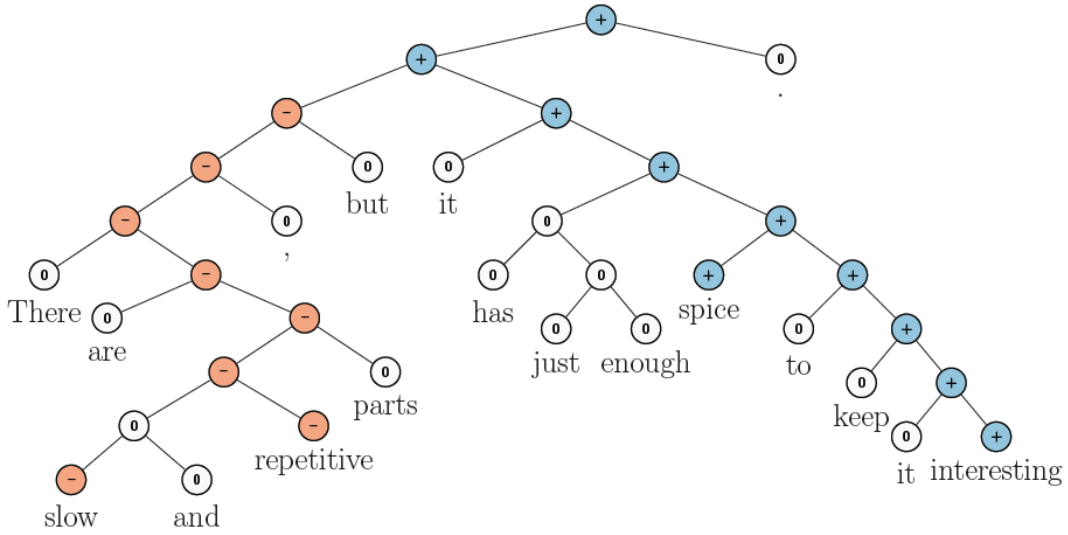
RNTN: Recursive Neural Tensor Network
(Socher et al., 2012)



- **同一のテンソルVと行列Wを合成時に使用**
  - パラメータの指数的増加の防止
- **テンソルによる合成**
  - 一つのテンソル層のスライスが、入力ベクトルに関するある特定の重み付けをした合成に対応
  - ２つの特徴ベクトル間の組み合わせを直接考慮

RNTN応用: 句・文の評価極性分類

(Socher et al., 2013)



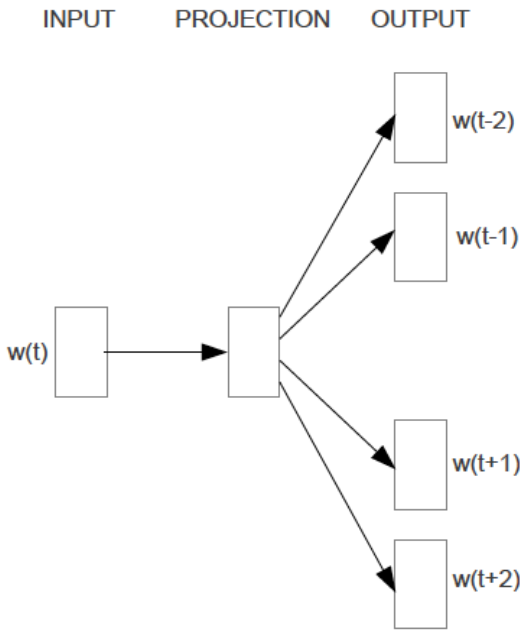
Model	Fine-grained	
	All	Root
NB	67.2	41.0
SVM	64.3	40.7
BiNB	71.0	41.9
VecAvg	73.3	32.7
RNN	79.0	43.2
MV-RNN	78.7	44.4
RNTN	**80.7**	**45.6**

- **タスク**
  - 単語と構文木の中間ノードに対して、極性付与
  - 5値: positive++, positive, neutral, negative, negative++
- **RNN, MV-RNNと比較して高い性能**
  - 極性の反転には、入力ベクトルの様々な組み合わせを考慮して合成することが有効

目次

- 言語処理分野におけるディープラーニング
- Deep NNに基づく構造予測
- 言語モデル・単語の分散表現
- Deep NNを用いた言語の構成性のモデル化
- 言語処理でディープアーキテクチャは必要か？
- まとめ

単語の分散表現を得る、Deep NNよりも優れたLog-linearモデルの出現



- **Log-linearモデル: Skip-gram (Mikolov et al., 2013)**
  - 隠れ層無し
  - ポテンシャル関数は、入力と出力の単語ベクトルの内積

$$p(w_O|w_I) = \frac{\exp(v'_{w_O} \cdot v_{w_I})}{\sum_{w=1}^{W} \exp(v'_w \cdot v_{w_I})}$$

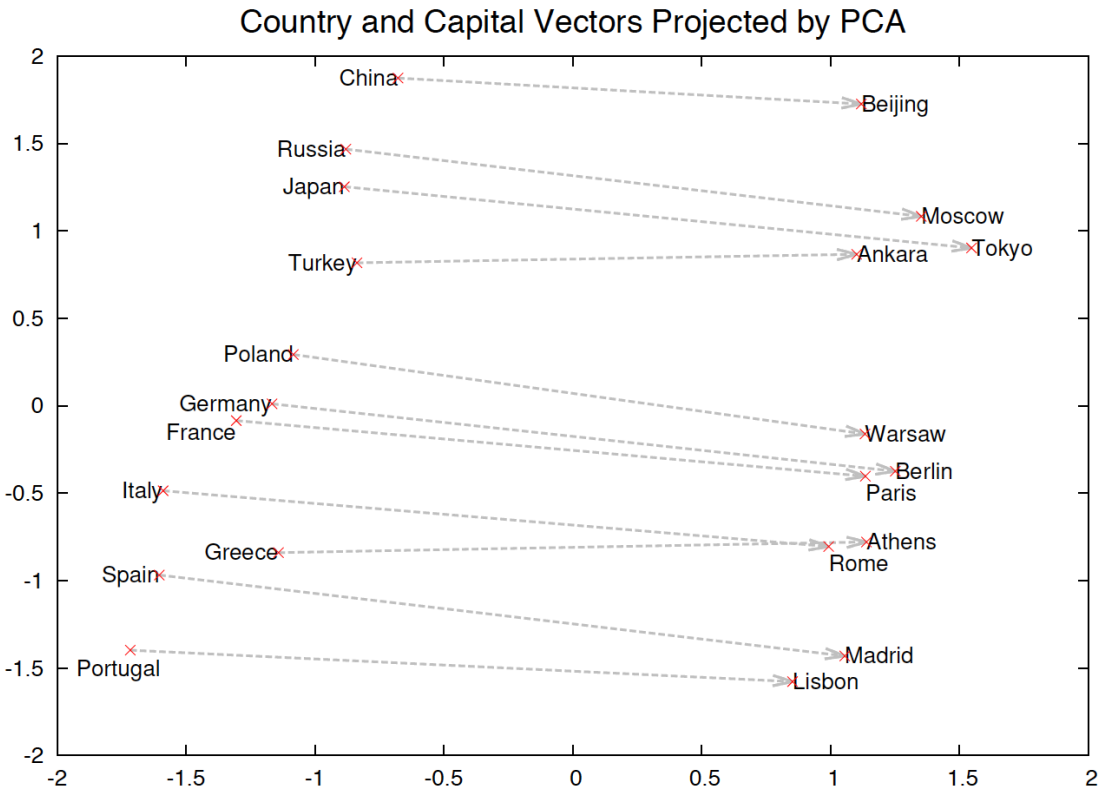
Skip-gram

Model Architecture	Semantic-Syntactic Word Relationship test set		MSR Word Relatedness Test Set [20]
	Semantic Accuracy [%]	Syntactic Accuracy [%]	
RNNLM	9	36	35
NNLM	23	53	47
CBOW	24	64	61
Skip-gram	55	59	56

国＋属性＝対応する名詞

各クエリと類似度の近いベクトル

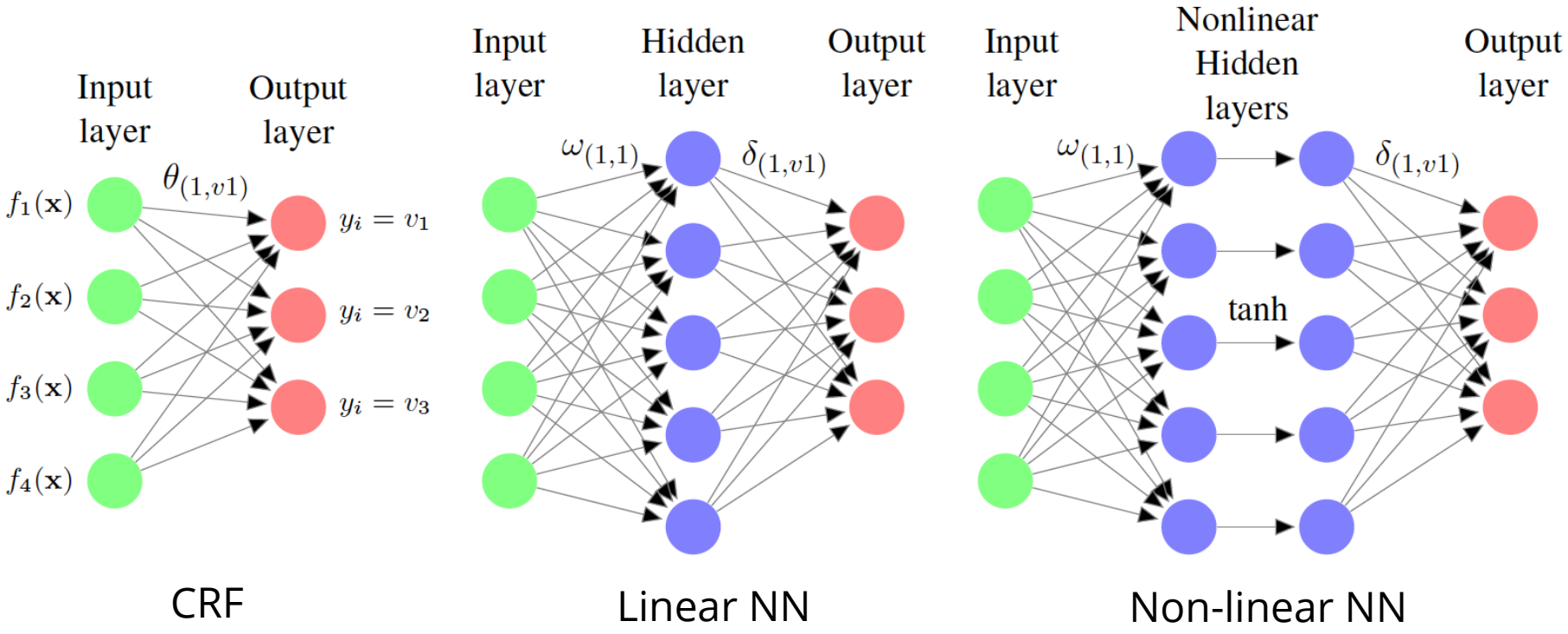
Czech + currency	Vietnam + capital	German + airlines	Russian + river	French + actress
koruna	Hanoi	airline Lufthansa	Moscow	Juliette Binoche
Check crown	Ho Chi Minh City	carrier Lufthansa	Volga River	Vanessa Paradis
Polish zolty	Viet Nam	flag carrier Lufthansa	upriver	Charlotte Gainsbourg
CTK	Vietnamese	Lufthansa	Russia	Cecile De



- vec(“ある国の首都”) – vec(“国”) = vec(“首都)
- 各国で類似したベクトルが得られている
- 単語の持つ意味がベクトルに反映されている

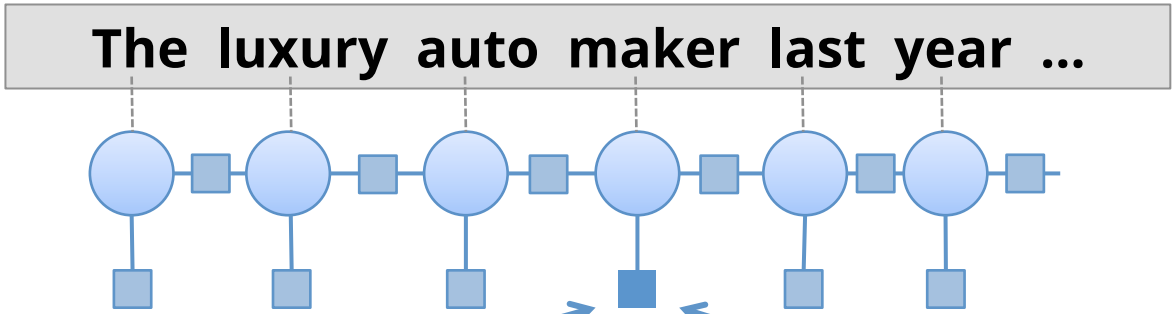
Deep NNのような非線形モデルは、従来手法と比較してどの程度優れているのか？

- **非線形アーキテクチャのNLPタスクにおける有効性を調査 (Wang and Manning 2013)**
  - Deep NN (2層、tanhなし、tanhあり) とConditional Random Fields (CRFs) (Lafferty+ 2001) を比較



CRF　　Linear NN　　Non-linear NN

CRFとDeep NNでのポテンシャル関数計算



$\Theta^{\mathrm{T}}\mathbf{f}(\mathbf{x})$

Θ

$\mathbf{f}(\mathbf{x})$ 0 0 0 1 0 0 1 0 ..... 1 1 0

-1:auto -1:NN +1:last +1:JJ

観測から得られる高次元離散ベクトル

CRF

$\Delta^{\mathrm{T}}\tanh(\Omega^{\mathrm{T}}\mathbf{f}(\mathbf{x}))$

Δ

tanh

Ω

低次元連続値ベクトル

luxury auto maker year last

Deep NN

複数タスクでのCRFとDeep NNとの性能比較
(Wang and Manning 2013)

特徴ベクトルが高次元離散ベクトルの場合

	CRF			多層NN		
	P	R	F1	P	R	F1
固有表現抽出 dev	90.9	90.4	**90.7**	89.3	89.7	89.5
固有表現抽出 test	85.4	84.7	**85.0**	83.3	83.9	83.6
固有表現抽出 out-of-domain 1	81.0	74.2	**77.4**	80.9	74.0	77.3
固有表現抽出 out-of-domain 2	72.5	74.5	**73.5**	71.1	74.1	72.6

従来手法の性能に劣る

特徴ベクトルが低次元連続値ベクトルの場合
(Collobert+ 2011 の word representationを利用)

	CRF			多層NN		
	P	R	F1	P	R	F1
固有表現抽出 dev	80.7	78.7	79.7	86.1	87.1	**86.6**
固有表現抽出 test	76.4	75.5	76.0	79.8	81.7	**80.7**
固有表現抽出 out-of-domain 1	71.5	71.1	71.3	75.8	74.1	**75.0**
固有表現抽出 out-of-domain 2	65.3	74.0	69.4	65.7	76.8	**70.8**

複数タスクでのCRFとDeep NNとの性能比較
(Wang and Manning 2013)

高次元離散ベクトルと低次元連続値ベクトルを組み合わせた場合

	固有表現抽出 dev	固有表現抽出 test	固有表現抽出 OOD1	固有表現抽出 OOD2
CRF 高次元離散	90.7	85.0	77.4	73.5
CRF 高次元離散+低次元連続	92.4	87.7	82.2	81.1
多層NN 低次元連続	86.6	80.7	75.0	70.8
多層NN 低次元連続＋高次元離散	91.9	87.1	81.2	79.7

双方のタイプの特徴ベクトルを組み合わせることで性能向上

言語処理におけるディープラーニング 現状のまとめ

- **Deep NNアーキテクチャ**
  - 言語モデルは一定の成果が出ている
  - 言語解析は従来手法（e.g. CRF）からの大幅な改善はない
- **分散表現 (Distributed Representation)**
  - 離散ベクトルとの組み合わせは有効
  - 意味的な類推は面白い結果が出つつある
- **表現学習 (Representation Learning)**
  - 言語処理タスクで必要な特徴は画像や音声と比較すると単純 ⇒ それほど旨味は無さそう？
    - 1単語 >>> 1ピクセル
- **言語の構成性のモデル化**
  - Recursive NNは強力なフレームワーク
  - 深い言語処理への応用の可能性を秘めている

言語処理におけるディープラーニング
今後の発展

- **良い分散表現の学習方法について**
  - 効率的かつ優れた表現を得るフレームワークの調査
  - 分散表現を用いた意味関係の類推
- **フレーズレベルの類義性判定**
  - A prevents B ⇔ A reduces the risk of B
  - 限定された品詞の組み合わせでは成果が出つつある (Tsubaki et al., 2013)
- **マルチモーダル**
  - 例：画像とテキスト

まとめ

- **言語処理分野におけるディープラーニング**
  - 構造予測: Multi-layer NN + Convolution
  - 言語モデル、分散表現: 多層NN, Recurrent NN
  - 言語の構成性の表現: Recursive NN
- **現状**
  - 性能面では従来の言語解析技術と比較して良い性能が得られているとは言い難い
  - 分散表現を用いた意味レベルの演算、構成的な意味計算は面白い結果が出つつある
- **今後**
  - 効率的かつ良質な分散表現の学習方法
  - フレーズレベルの意味関係の推論
  - マルチモーダル