

# Neural CRF Parsing

Greg Durrett, Dan Klein  
ACL 2015

統数研 持橋大地

[daichi@ism.ac.jp](mailto:daichi@ism.ac.jp)

最先端NLP勉強会 2015-8-29 (土)

# おまけ

Amazon.co.jp : Probability: Theory and Examples (Cambridge Series in Statistical and Probabilistic Mathematics): Rick Durrett: 洋書

http://www.amazon.co.jp/Probability-Cambridge-Statistical-Probabilistic-Mathematics/dp/0521717171

なか見!検索

**Probability: Theory and Examples**  
Cambridge Series in Statistical and Probabilistic Mathematics  
Fourth Edition  
Rick Durrett

**Probability: Theory and Examples (Cambridge Series in Statistical and Probabilistic Mathematics) (英語) ハードカバー**  
- 2010/8/30  
Rick Durrett (著)  
カスタマーレビューを書きませんか?

すべての 5 フォーマットおよびエディションを表示する

Kindle版 ¥ 6,527	ハードカバー ¥ 11,410
--------------------	--------------------

今すぐお読みいただけます: **無料アプリ**

¥ 9,000 より 3 中古品の出品  
¥ 9,800 より 22 新品

Amazon Student 会員なら、この商品は+10%Amazonポイント還元(Amazonマーケットプレイスでのご注文は対象外)。無料体験でもれなくポイント1,000円分プレゼント  
キャンペーン実施中。

Would you like to see this page in English?  
Click here.

シェアする

**新品**  
¥ 11,410  
参考価格: ¥ 11,532  
OFF: ¥ 122 (1%)

通常配送無料 詳細

通常2~3週間以内に発送します。在庫状況について  
この商品は、Amazon.co.jp が販売、発送します。  
ギフトラッピングを利用できます。

数量: 1

カートに入れる

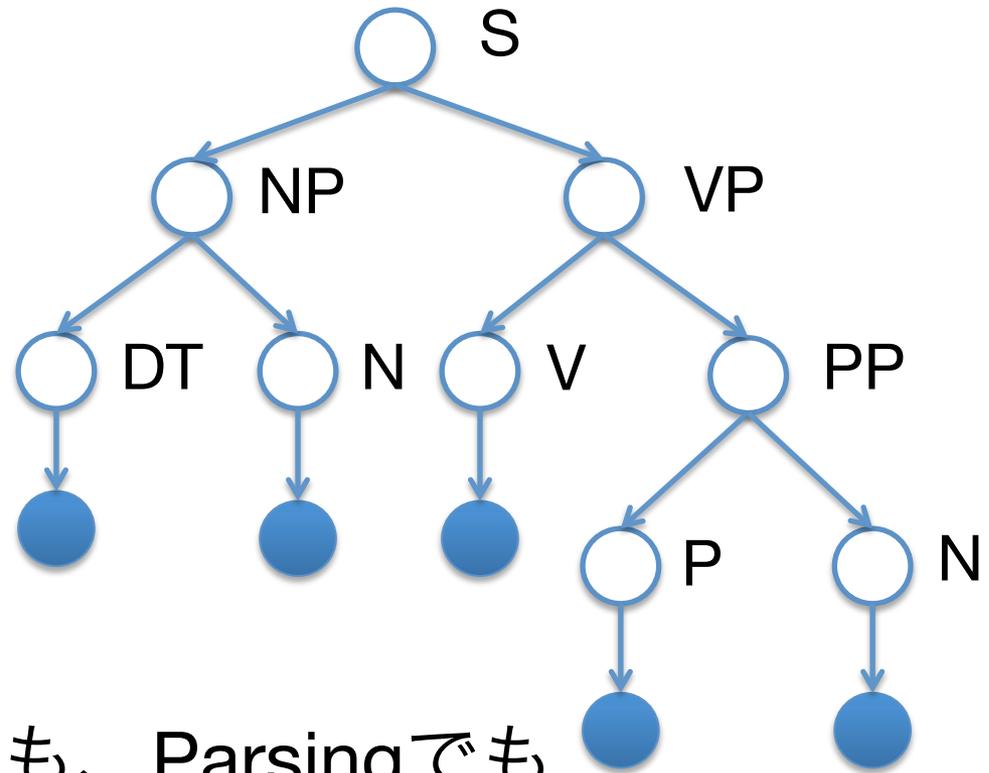
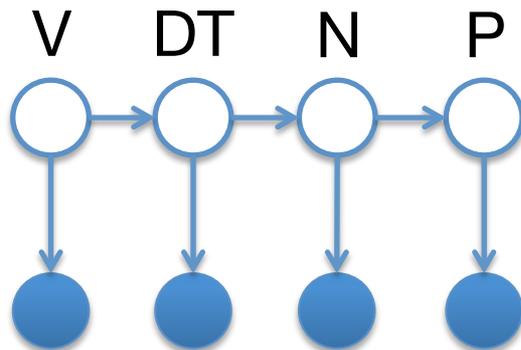
1-Clickで注文する場合は、サインインをしてください。

## ● の人の子供

# 論文の概要

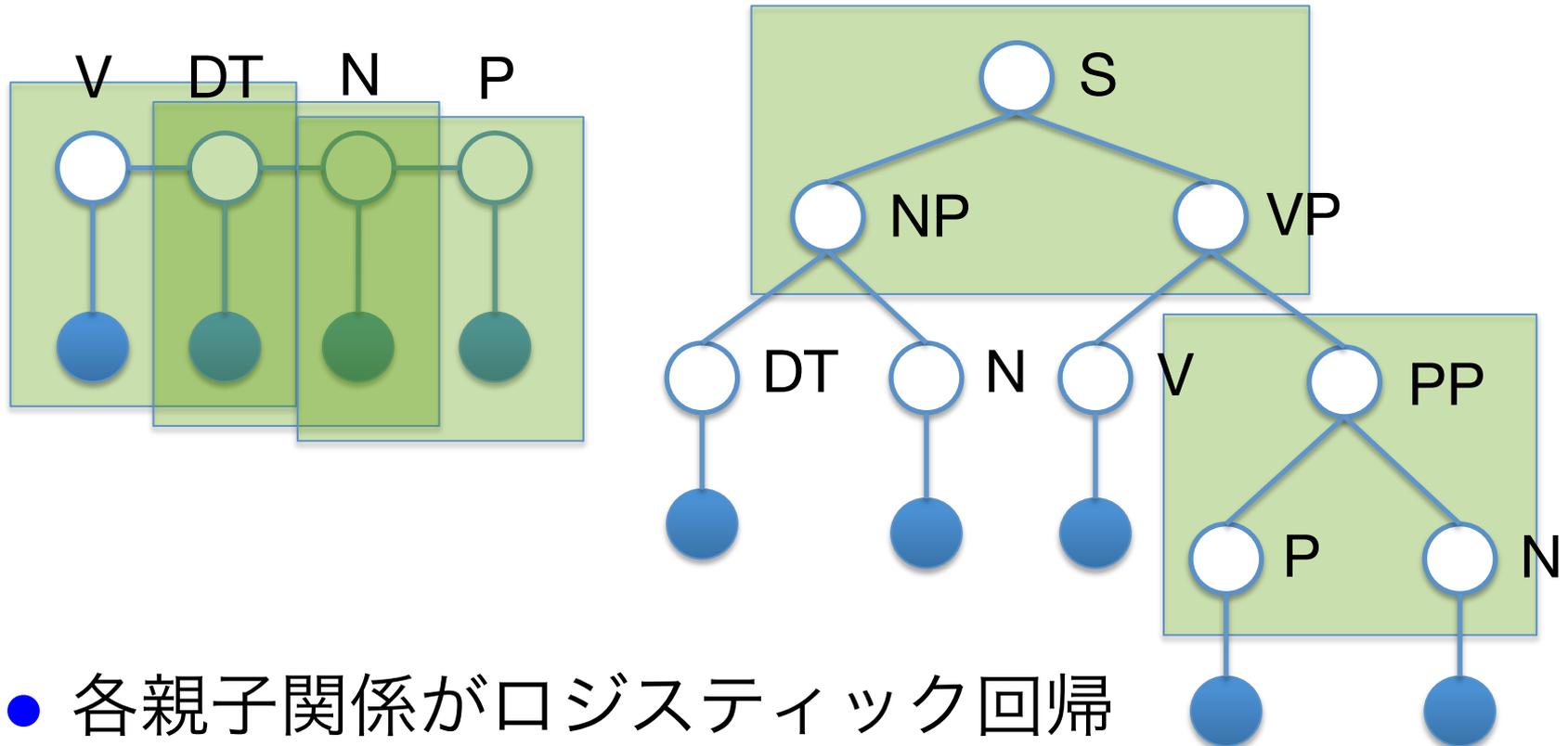
- Hall (2014) “Less grammar, More features” の連続化
- 「こんな感じの語で始まるとPP」「こんな感じの語で終わるとNP」などをモデル化できる
  - 単語がExactに一致している必要はない
  - 既知の単語埋め込みをもとに、スコアに変換する行列を学習
- 連続→離散の連繋、通常のNNと異なりCKYなどこれまでのアルゴリズムが普通に使える

# Parsing with CRFs



- Linear-chain CRFでも、Parsingでも動的計画法の仕組みは一緒 (Forward-Backward)

# Parsing with CRFs (2)



- 各親子関係がロジスティック回帰
  - CRF=ロジスティック回帰のMarkovモデル
  - $p(\text{ルール}) \propto \exp(\phi(\mathbf{w}, \text{ルール}))$      $\phi$ :スコアを返す関数

# 通常CRFスコア

- PCFGの確率は、導出確率の積

$$\begin{aligned} p(T|\mathbf{w}) &= \prod_r p(r|\text{parent}(r)) \\ &= \exp\left(\sum_r \log p(r|\text{parent}(r))\right) \end{aligned}$$

- CRFの場合: スコア関数  $\phi$  を任意に設定

$$p(T|\mathbf{w}) = \exp\left(\sum_r \phi(\mathbf{w}, r)\right)$$

- $\phi = \text{重み} \times \delta(\text{ルール})$  とするのが最も簡単な関数
- $\phi = \log p(r|\text{parent}(r))$  がPCFGの場合



# 通常CRFスコア $\phi$ の計算

- 素性 $k$ がルール $r$ と共起する重み =

$$f_s(k) \cdot f_o(r) \cdot w_{kr}$$

- 発火する素性およびルールは複数あるので、

$$\sum_{k \in f(s)} \sum_{r \in f(o)} f_s(k) \cdot f_o(r) \cdot w_{kr}$$

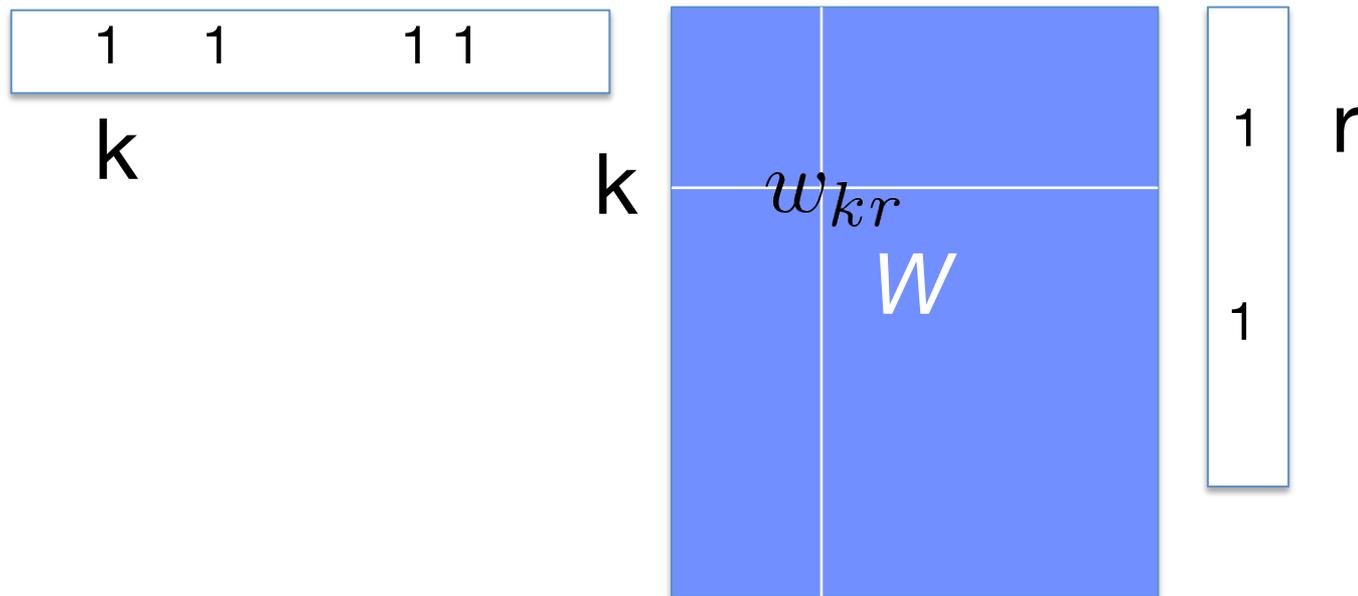
と書ける

# 通常のCRFスコア $\phi$ の計算 (2)

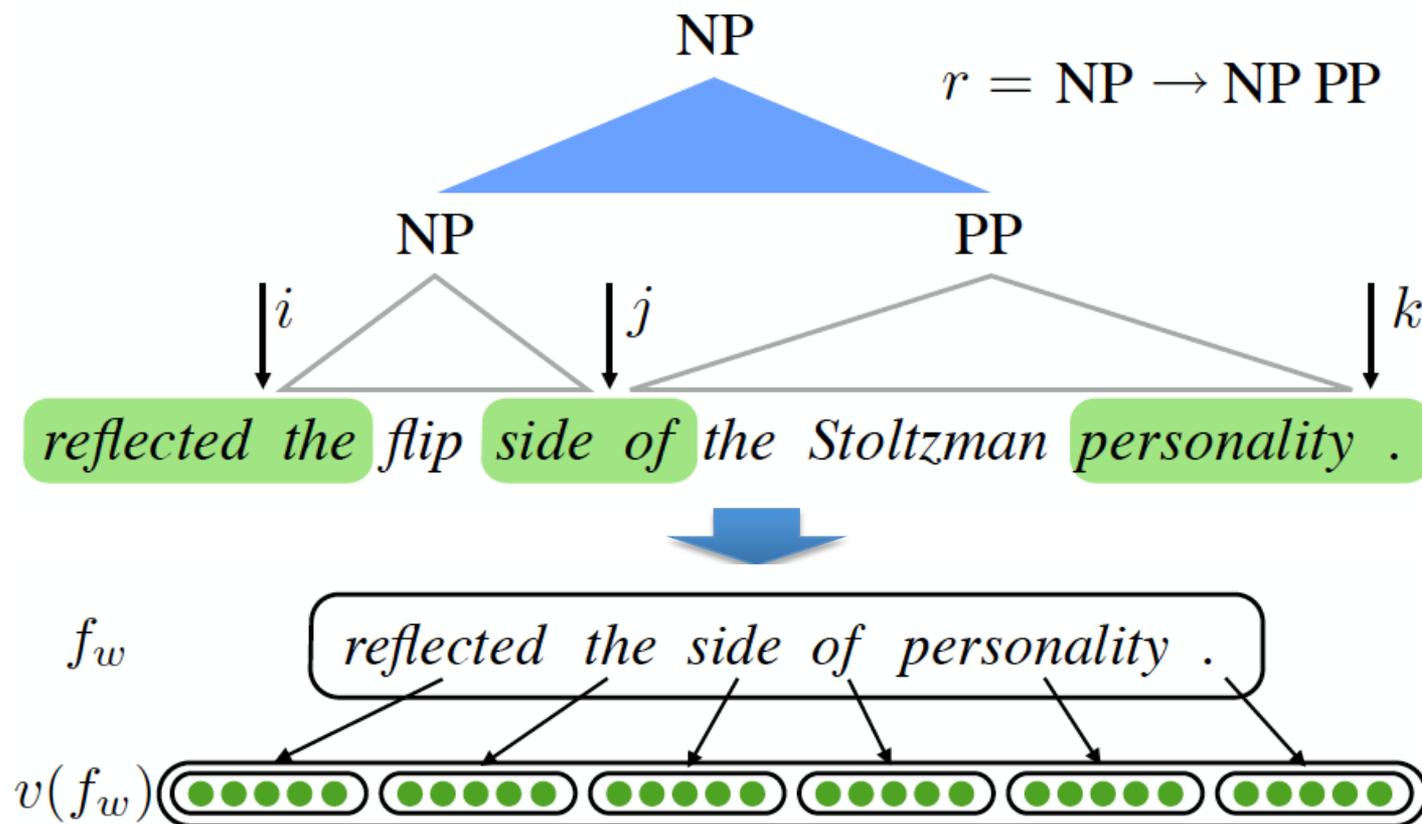
- 行列形式で書くと、

$$\phi(\mathbf{w}, r) = f_s(\mathbf{w})^T W f_o(r)$$

$$\phi(\mathbf{w}, r) =$$



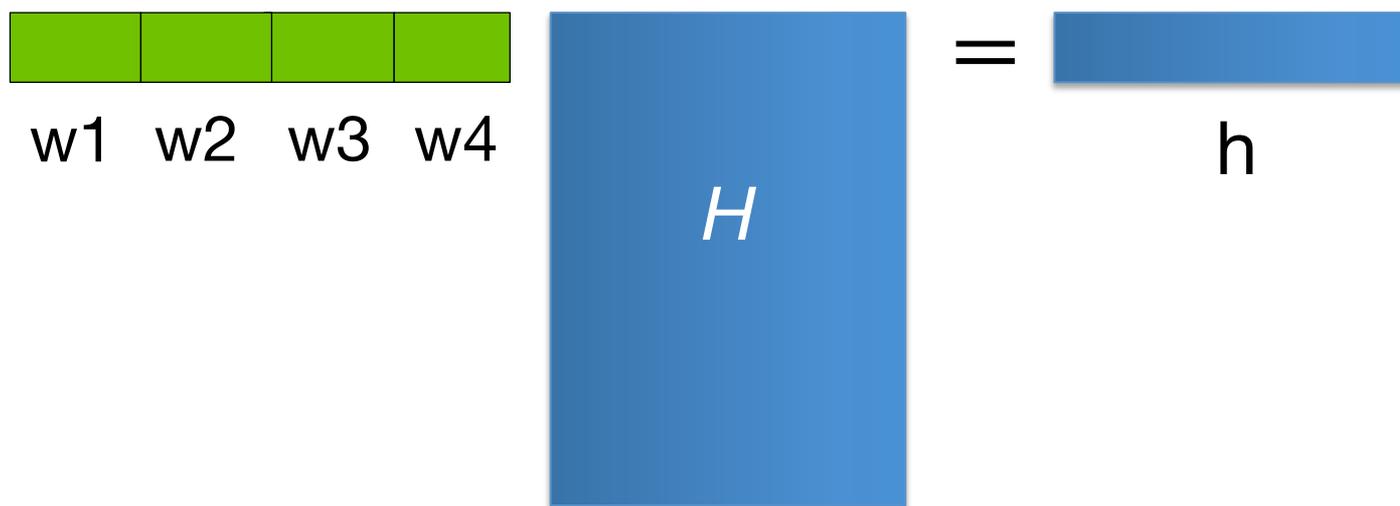
# 単語ベクトルを使った連続化



- 素性に疎な0/1ベクトルを使うかわりに、単語ベクトルを連結して入力

## 単語ベクトルを使った連続化 (2)

- 単語ベクトルの連結に行列 $H$ を掛けて、(連続な)隠れベクトル $h$ にする



- $h$ をReLUで非線形化したものが素性ベクトル

# 単語ベクトルによる連続化 (3)

- 最終的なポテンシャル関数の形:

$$\phi(\mathbf{w}, r) = \text{ReLU} \left( \begin{array}{c} \begin{array}{cccc} \color{green}{\square} & \color{green}{\square} & \color{green}{\square} & \color{green}{\square} \\ w_1 & w_2 & w_3 & w_4 \end{array} & \begin{array}{c} \color{blue}{\square} \\ H \end{array} \end{array} \right) \begin{array}{c} \color{blue}{\square} \\ W \end{array} \begin{array}{c} \color{white}{\square} \\ f_o \end{array}$$

- $W, H$ に関する(劣)微分は通常通り計算できる
- あとはCKY!

# 連続素性+離散素性

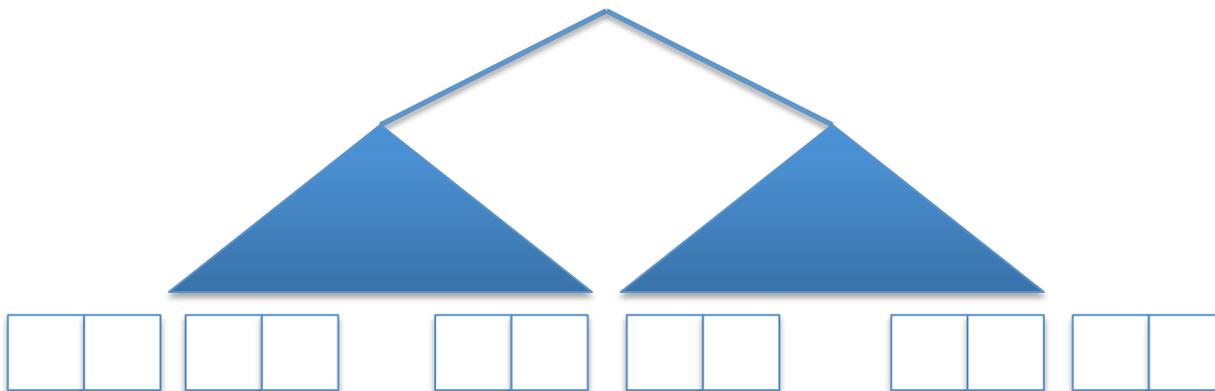
- 2種類のポテンシャルを足し合わせればよい

$$\begin{aligned}\phi(\mathbf{w}, r, s; W_1, H, W_2) &= \phi_{\text{sparse}}(\mathbf{w}, r, s; W_1) \\ &\quad + \phi_{\text{neural}}(\mathbf{w}, r, s; H, W_2)\end{aligned}$$

- 「特定の単語」が予測に効く可能性がある

# 文法と素性

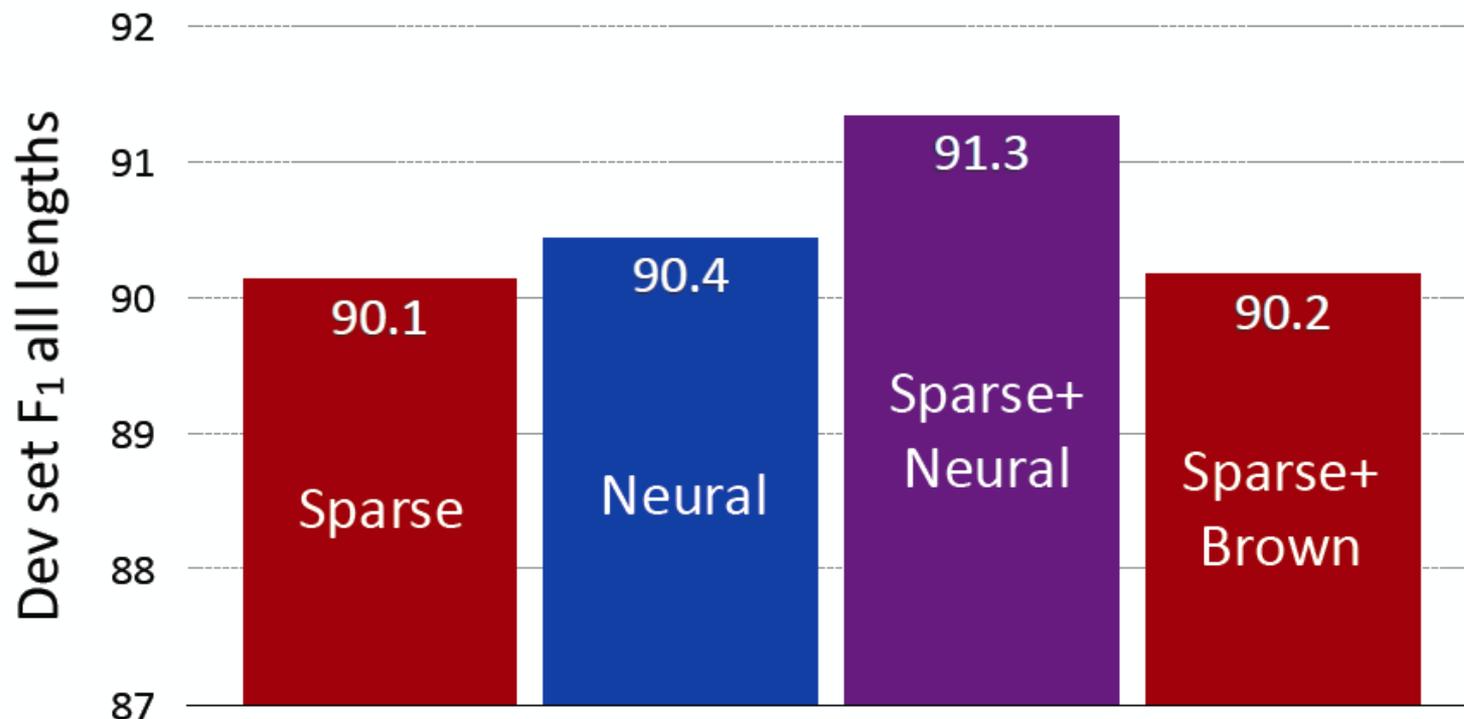
- 探索空間を複雑にしないため、文法は最小限
  - 英語では、マルコフ化なし
  - 英語以外では、親だけマルコフ化 (兄弟はなし)
- 素性はHall+(2014)と同じ
  - Preterminalでは、自分+前後5個の単語
  - Nonterminalでは、Spanの境界 $\pm 2$ =全部で12語



# 実験結果



## Results: English Treebank (Dev)

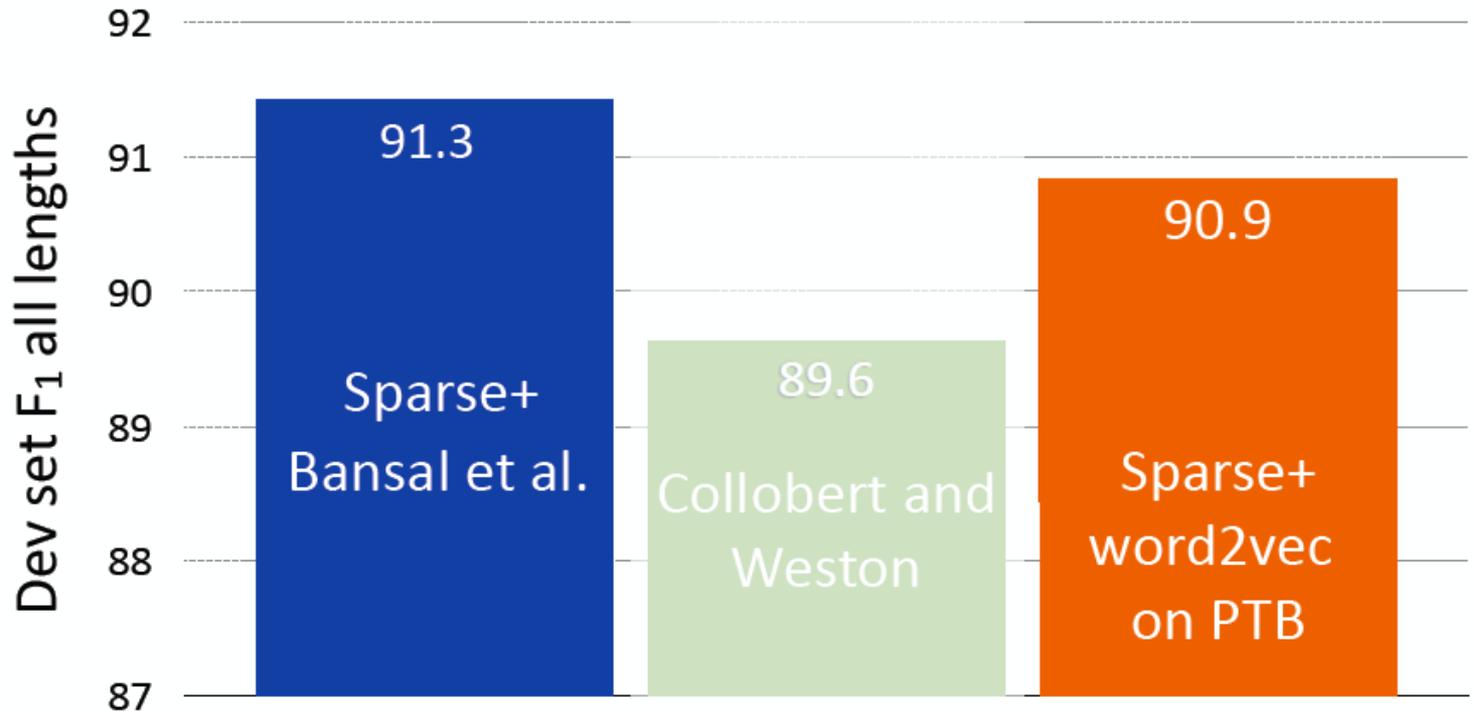


- 連続のみでも、離散素性の性能を超えている

# 使用した単語ベクトル



## Word Vectors

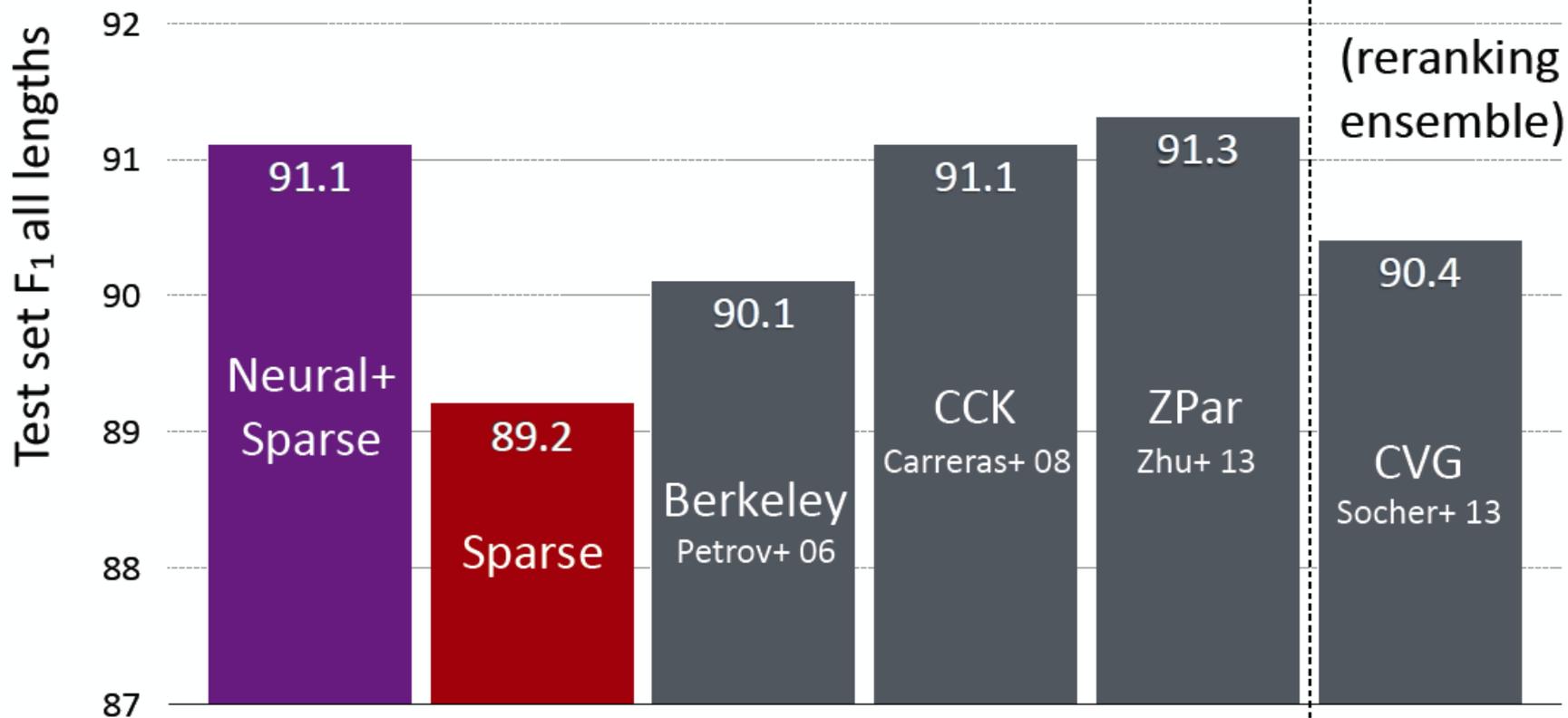


- Bansal+(2014)の係り受け用の埋め込みが高性能
  - 埋め込み自体を学習しても、性能は上がらなかった

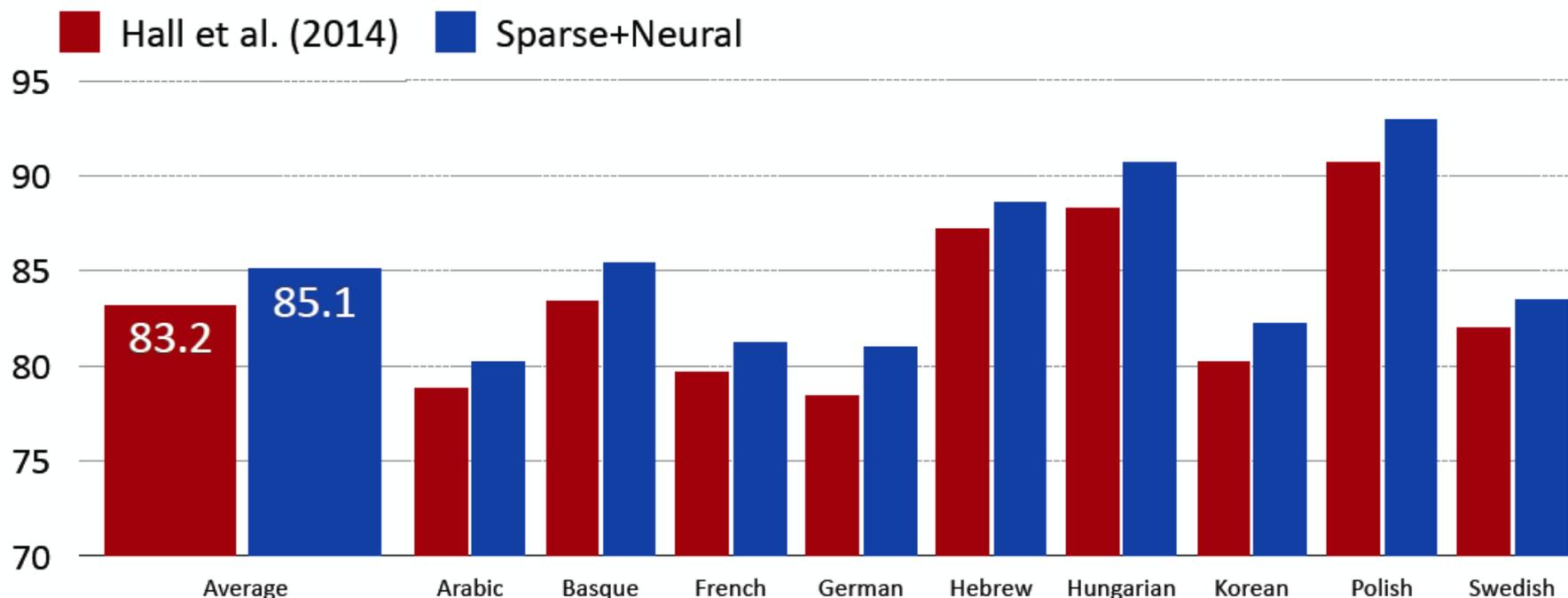
# WSJテストデータ



## Results: English Treebank (Test)



# 他の言語



- すべての言語で、Hall+(2014)より高性能
  - 連続素性により汎化性能が高いため、小データでも有効

# まとめ

- CRFで用いる素性を連続化
  - 既知の単語埋め込みをもとに、それを潜在素性に変換する行列 $H$ を学習
  - 「こんな感じの単語」という概念が、離散の枠組の中で扱える
  - Parsingの話に限らない一般的な話
- 通常のDNNと異なり、CRFや動的計画法などコンピュータサイエンスのアルゴリズムがそのまま使える
- Single Parserとして現在最高精度  
<http://nlp.cs.berkeley.edu/projects/neuralcrf.shtml>