自然言語処理プログラミング勉強会 11
構造化パーセプトロン

Graham Neubig
奈良先端科学技術大学院大学 (NAIST)

予測問題

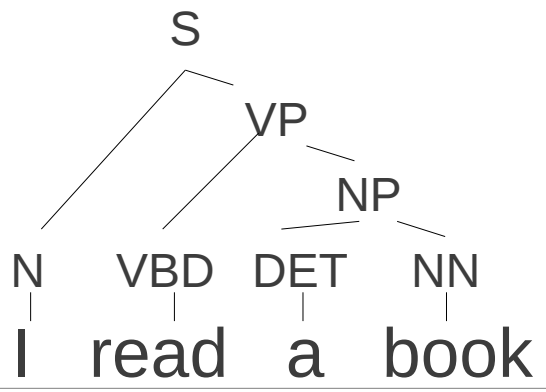
x が与えられた時　　y を予測する

本のレビュー	「良い」評価なのか？
Oh, man I love this book!	yes
This book is so boring...	no

2値予測
(選択肢が2つ)

ツイート	書かれた言語
On the way to the park!	English
公園に行くなう!	Japanese

多クラス予測
(選択肢が数個)

文	構文木
I read a book	

構造化予測
(選択肢が膨大)

予測問題

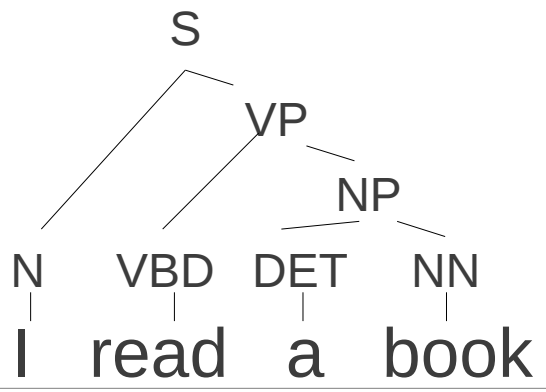
x が与えられた時　　y を予測する

本のレビュー	「良い」評価なのか？
Oh, man I love this book!	yes
This book is so boring...	no

2値予測
(選択肢が2つ)

ツイート	書かれた言語
On the way to the park!	English
公園に行くなう!	Japanese

多クラス予測
(選択肢が数個)

文	構文木
I read a book	

自然言語処理のほとんど!

構造化予測
(選択肢が膨大)

今まで勉強した予測手法

2値分類器

パーセプトロン , SVM,
ニューラルネット

多くの素性

2値予測

生成モデル

HMM 品詞推定
PCFG 構文解析

最尤推定による確率推定

構造予測

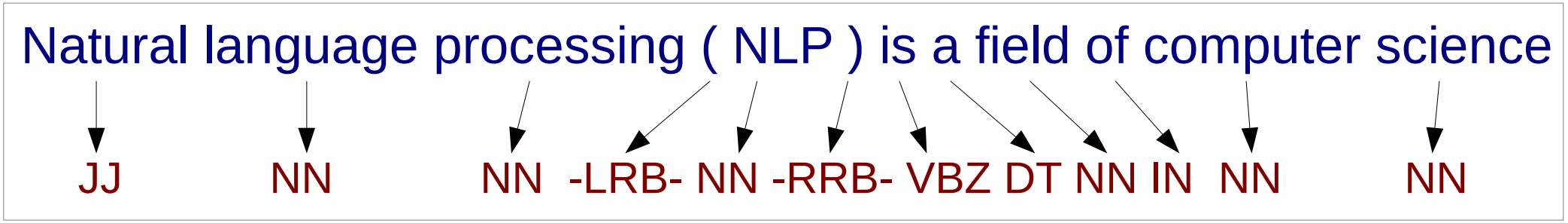
構造化パーセプトロン →
構造予測に対して、多くの素性が利用可能!

構造化パーセプトロンの利用例

- HMM 品詞推定
  Collins “Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms” ACL02
- 構文解析
  Huang+ “Forest Reranking: Discriminative Parsing with Non-Local Features” ACL08
- 機械翻訳
  Liang+ “An End-to-End Discriminative Approach to Machine Translation” ACL06
  (Neubig+ “Inducing a Discriminative Parser for Machine Translation Reordering”, EMNLP12, ステマ :))
- 識別言語モデル
  Roark+ “Discriminative Language Modeling with Conditional Random Fields and the Perceptron Algorithm” ACL04

例：品詞推定

- 文 X が与えられた時の品詞列 Y を予測する



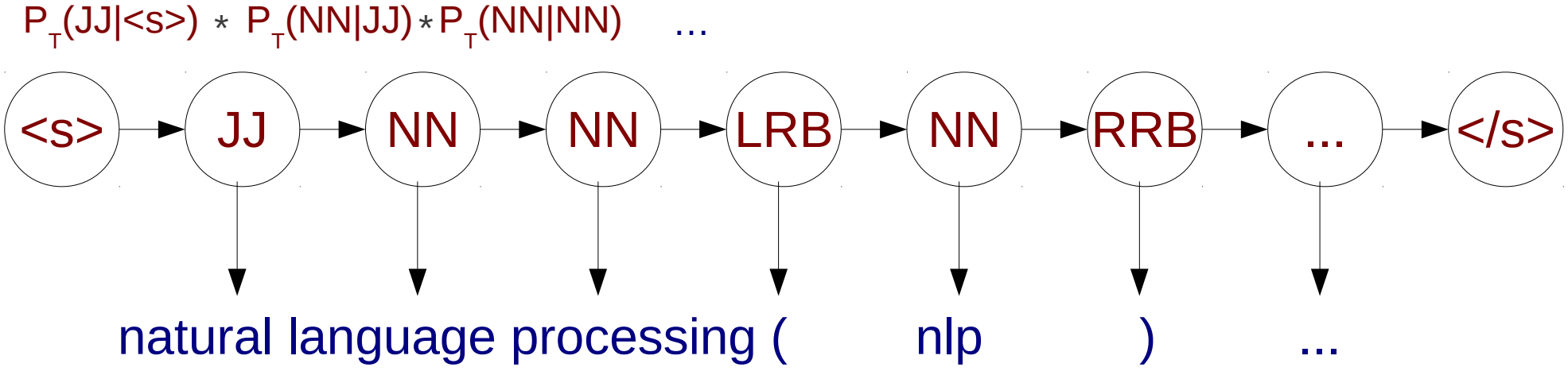
- 「構造予測」に分類される

復習：品詞推定のための (HMM)

- 品詞→品詞の遷移確率
  - 2-gram モデルとほぼ一緒
- 品詞→単語の生成確率

$$P(Y) \approx \prod_{i=1}^{I+1} P_T(y_i|y_{i-1})$$

$$P(X|Y) \approx \prod_{1}^{I} P_E(x_i|y_i)$$



素性はなぜ必要？

- HMM なら、確率の兼ね合いなどに配慮が必要
- 素性なら新しいアイデアをどんどん試せる
  - 文中に大文字として現れるものは名詞が多い？
    →「名詞＋大文字」素性の追加
  - 「 -ed 」や「 -ing 」で終わる単語は動詞が多い？
    →「動詞 +-ed 」「動詞 +-ing 」素性の追加

素性が使えるように HMM を変形

通常の HMM

$$P(X,Y)=\prod_{1}^{I} P_E(x_i|y_i)\prod_{i=1}^{I+1} P_T(y_i|y_{i-1})$$

素性が使えるように HMM を変形

通常の HMM

$$P(X,Y)=\prod_{1}^{I} P_E(x_i|y_i)\prod_{i=1}^{I+1} P_T(y_i|y_{i-1})$$

対数尤度

$$\log P(X,Y)=\sum_{1}^{I} \log P_E(x_i|y_i)\sum_{i=1}^{I+1} \log P_T(y_i|y_{i-1})$$

素性が使えるように HMM を変形

通常の HMM

$$P(X,Y)=\prod_1^{I} P_E(x_i|y_i)\prod_{i=1}^{I+1} P_T(y_i|y_{i-1})$$

対数尤度

$$\log P(X,Y)=\sum_1^{I} \log P_E(x_i|y_i)\sum_{i=1}^{I+1} \log P_T(y_i|y_{i-1})$$

スコア

$$S(X,Y)=\sum_1^{I} w_{E,y_i,x_i}\sum_{i=1}^{I+1} w_{E,y_{i-1},y_i}$$

素性が使えるように HMM を変形

通常の HMM $$P(X,Y)=\prod_1^{I} P_E(x_i|y_i)\prod_{i=1}^{I+1} P_T(y_i|y_{i-1})$$

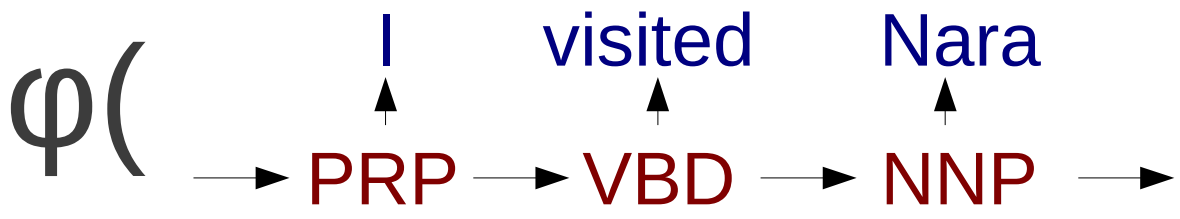
対数尤度 $$\log P(X,Y)=\sum_1^{I} \log P_E(x_i|y_i)\sum_{i=1}^{I+1} \log P_T(y_i|y_{i-1})$$

スコア $$S(X,Y)=\sum_1^{I} w_{E,y_i,x_i}\sum_{i=1}^{I+1} w_{E,y_{i-1},y_i}$$

$w_{E,y_i,x_i}=\log P_E(x_i|y_i)$　$w_{T,y_{i-1},y_i}=\log P_T(y_i|y_{i-1})$　なら

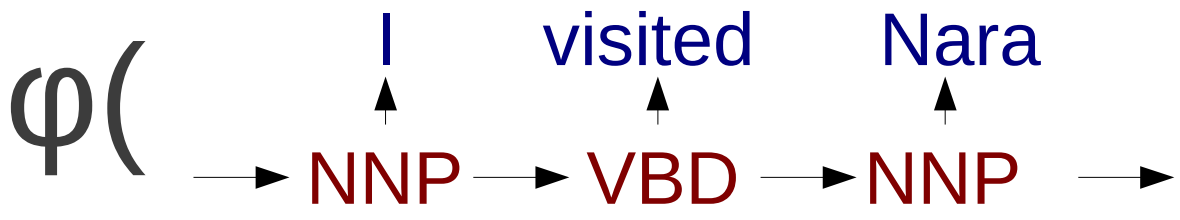
$\log P(X,Y) = S(X,Y)$　が成り立つ

例：



$\varphi_{T,<S>,PRP}(X,Y_1) = 1$ $\varphi_{T,PRP,VBD}(X,Y_1) = 1$ $\varphi_{T,VBD,NNP}(X,Y_1) = 1$ $\varphi_{T,NNP,</S>}(X,Y_1) = 1$

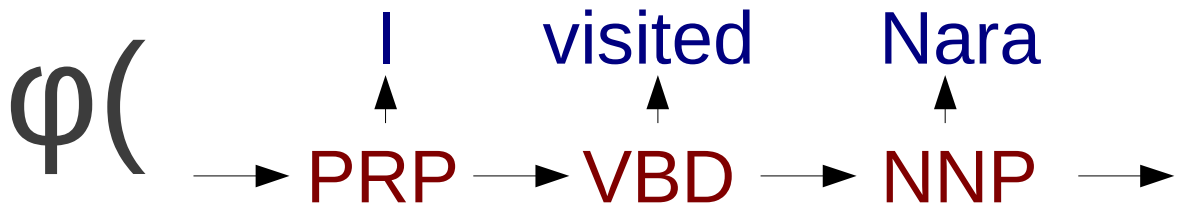
$\varphi_{E,PRP,"I"}(X,Y_1) = 1$ $\varphi_{E,VBD,"visited"}(X,Y_1) = 1$ $\varphi_{E,NNP,"Nara"}(X,Y_1) = 1$



$\varphi_{T,<S>,NNP}(X,Y_1) = 1$ $\varphi_{T,NNP,VBD}(X,Y_1) = 1$ $\varphi_{T,VBD,NNP}(X,Y_1) = 1$ $\varphi_{T,NNP,</S>}(X,Y_1) = 1$

$\varphi_{E,NNP,"I"}(X,Y_1) = 1$ $\varphi_{E,VBD,"visited"}(X,Y_1) = 1$ $\varphi_{E,NNP,"Nara"}(X,Y_1) = 1$

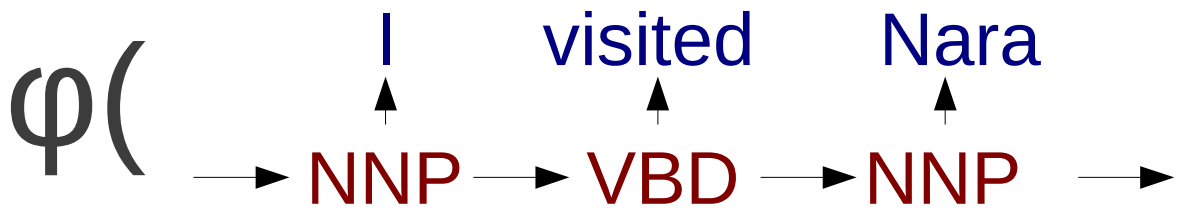
例：



$\varphi_{T,<S>,PRP}(X,Y_1) = 1$ $\varphi_{T,PRP,VBD}(X,Y_1) = 1$ $\varphi_{T,VBD,NNP}(X,Y_1) = 1$ $\varphi_{T,NNP,</S>}(X,Y_1) = 1$

$\varphi_{E,PRP,"I"}(X,Y_1) = 1$ $\varphi_{E,VBD,"visited"}(X,Y_1) = 1$ $\varphi_{E,NNP,"Nara"}(X,Y_1) = 1$

$\varphi_{CAPS,PRP}(X,Y_1) = 1$ $\varphi_{CAPS,NNP}(X,Y_1) = 1$

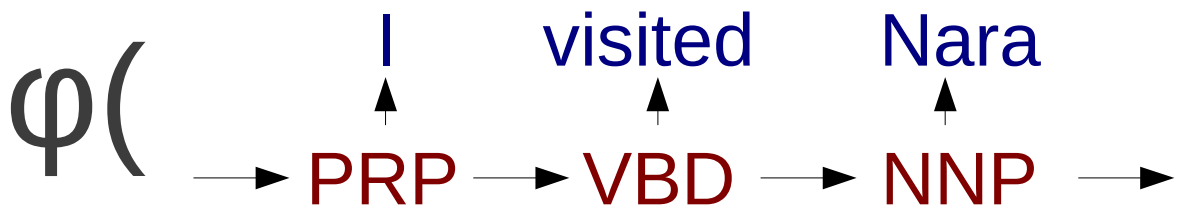


$\varphi_{T,<S>,NNP}(X,Y_1) = 1$ $\varphi_{T,NNP,VBD}(X,Y_1) = 1$ $\varphi_{T,VBD,NNP}(X,Y_1) = 1$ $\varphi_{T,NNP,</S>}(X,Y_1) = 1$

$\varphi_{E,NNP,"I"}(X,Y_1) = 1$ $\varphi_{E,VBD,"visited"}(X,Y_1) = 1$ $\varphi_{E,NNP,"Nara"}(X,Y_1) = 1$

$\varphi_{CAPS,NNP}(X,Y_1) = 2$

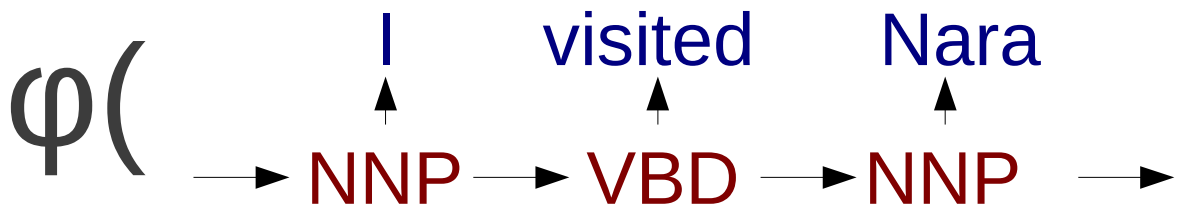
例：



$\varphi_{T,<S>,PRP}(X,Y_1) = 1$ $\varphi_{T,PRP,VBD}(X,Y_1) = 1$ $\varphi_{T,VBD,NNP}(X,Y_1) = 1$ $\varphi_{T,NNP,</S>}(X,Y_1) = 1$

$\varphi_{E,PRP,"I"}(X,Y_1) = 1$ $\varphi_{E,VBD,"visited"}(X,Y_1) = 1$ $\varphi_{E,NNP,"Nara"}(X,Y_1) = 1$

$\varphi_{CAPS,PRP}(X,Y_1) = 1$ $\varphi_{CAPS,NNP}(X,Y_1) = 1$ $\varphi_{SUF,VBD,"...ed"}(X,Y_1) = 1$



$\varphi_{T,<S>,NNP}(X,Y_1) = 1$ $\varphi_{T,NNP,VBD}(X,Y_1) = 1$ $\varphi_{T,VBD,NNP}(X,Y_1) = 1$ $\varphi_{T,NNP,</S>}(X,Y_1) = 1$

$\varphi_{E,NNP,"I"}(X,Y_1) = 1$ $\varphi_{E,VBD,"visited"}(X,Y_1) = 1$ $\varphi_{E,NNP,"Nara"}(X,Y_1) = 1$

$\varphi_{CAPS,NNP}(X,Y_1) = 2$ $\varphi_{SUF,VBD,"...ed"}(X,Y_1) = 1$

最適解の探索

- 以下の式に従って最適解を見つけたい

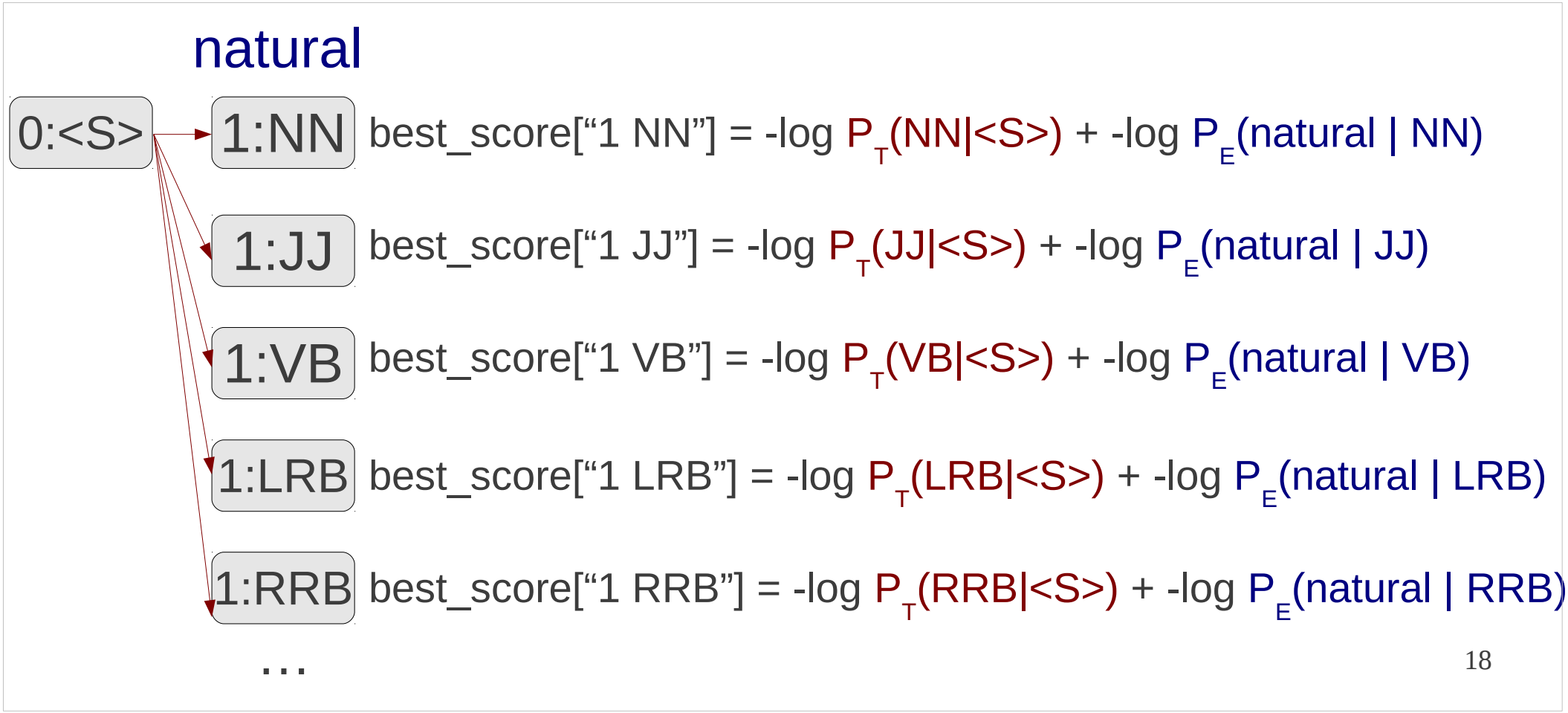
$$\hat{Y} = \mathrm{argmax}_Y \sum_i w_i \phi_i(X, Y)$$

復習： HMM のビタビ探索

- 前向きステップ：各ノードへたどる確率の計算
  - 負の対数尤度がもっとも低くなるパス
- 後ろ向きステップ：パスの復元
  - 単語分割とほとんど同じ

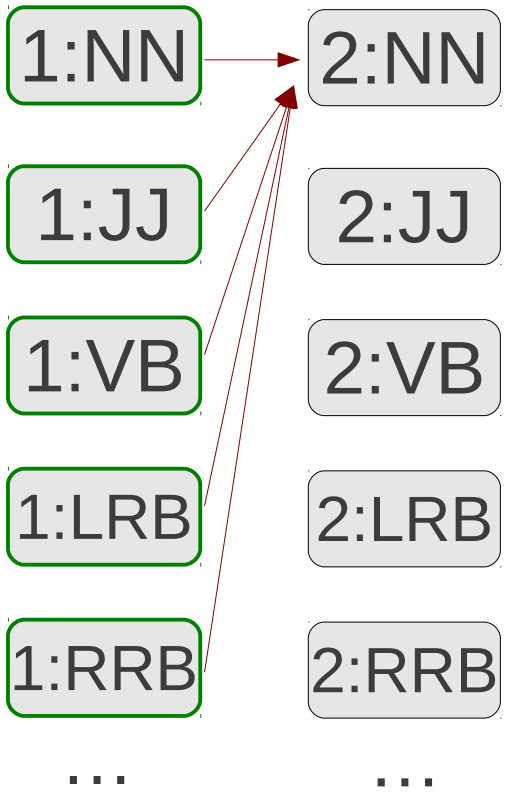
前向きステップ：文頭

- 文頭記号 <S> から 1 単語目への遷移と 1 単語目の生成の確率



前向きステップ：中間

- 前の品詞を全部比べて、これまでのパス、遷移、生成を全て考慮した最短パスを利用

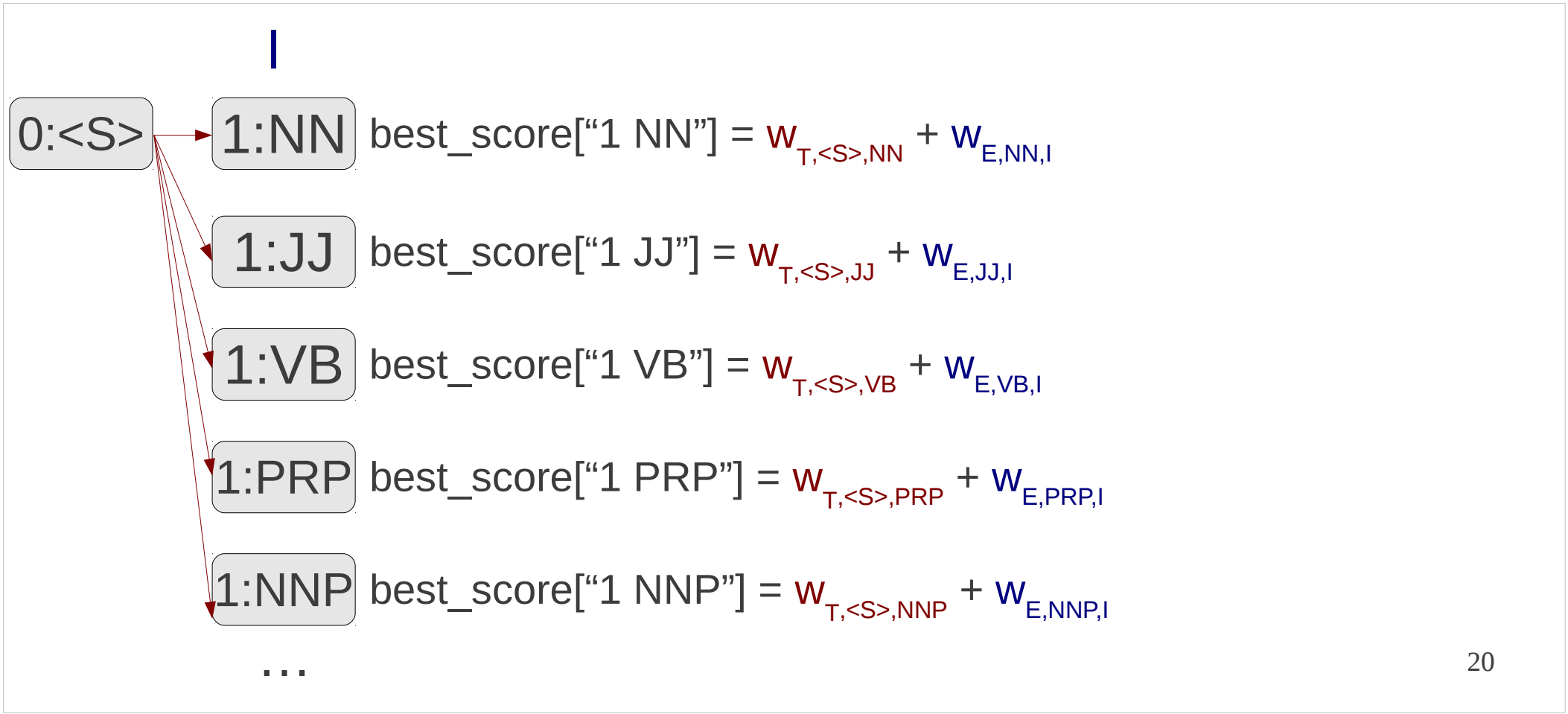


best_score[“2 NN”] = min(
best_score[“1 NN”] + -log P_T(NN|NN) + -log P_E(language | NN),
best_score[“1 JJ”] + -log P_T(NN|JJ) + -log P_E(language | NN),
best_score[“1 VB”] + -log P_T(NN|VB) + -log P_E(language | NN),
best_score[“1 LRB”] + -log P_T(NN|LRB) + -log P_E(language | NN),
best_score[“1 RRB”] + -log P_T(NN|RRB) + -log P_E(language | NN),
...
)

best_score[“2 JJ”] = min(
best_score[“1 NN”] + -log P_T(JJ|NN) + -log P_E(language | JJ),
best_score[“1 JJ”] + -log P_T(JJ|JJ) + -log P_E(language | JJ),
best_score[“1 VB”] + -log P_T(JJ|VB) + -log P_E(language | JJ),
...

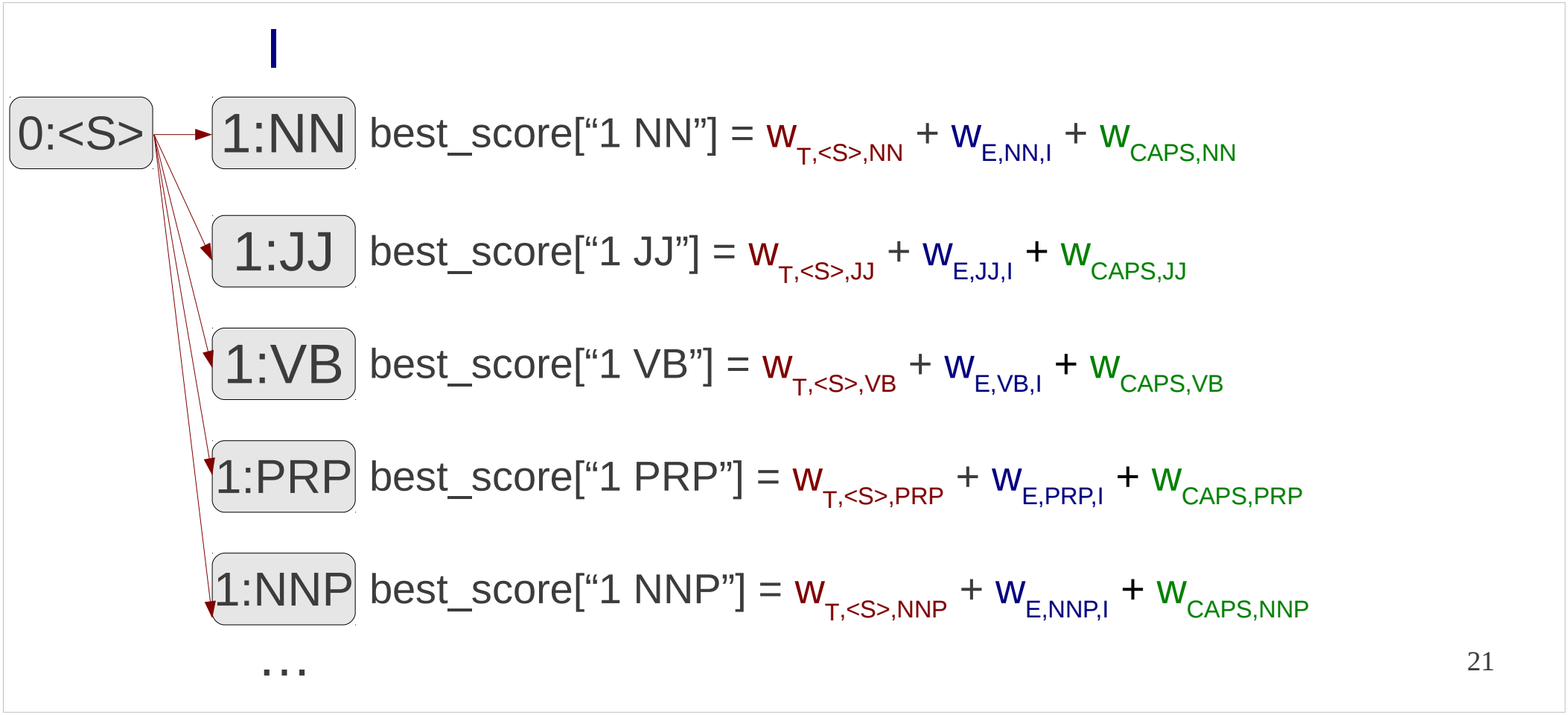
素性を使った HMM ビタビ

- 確率と同じように素性を利用



素性を使った HMM ビタビ

- 他の素性も導入可能



構造化パーセプトロンの学習

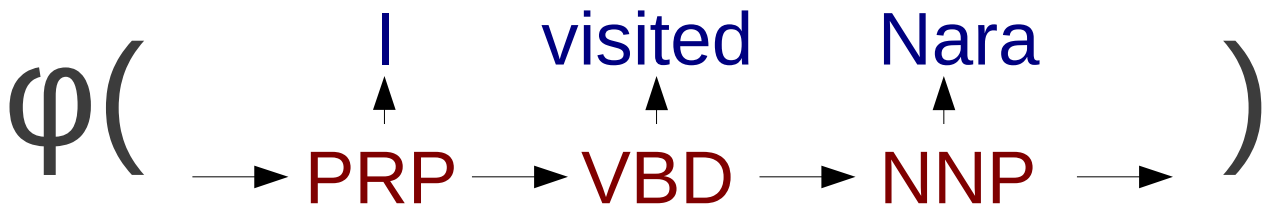
- パーセプトロンアルゴリズムの重み更新
- 誤りの場合：

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + y\,\boldsymbol{\phi}\left(x\right)$$

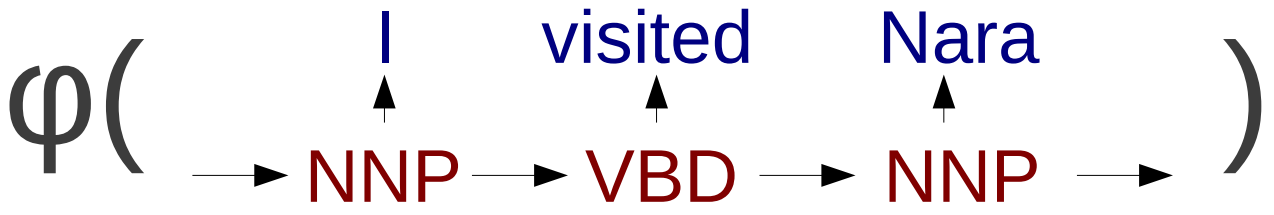
- 重みが：
  正例に対して大きくなる
  負例に対して小さくなる　ように更新
- 構造化パーセプトロンの「正例」と「負例」とは？

構造化パーセプトロンの学習

- 正例：正しいタグ列に対する素性ベクトル

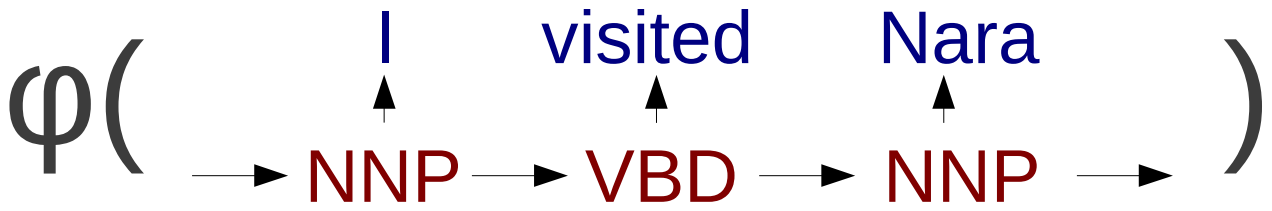


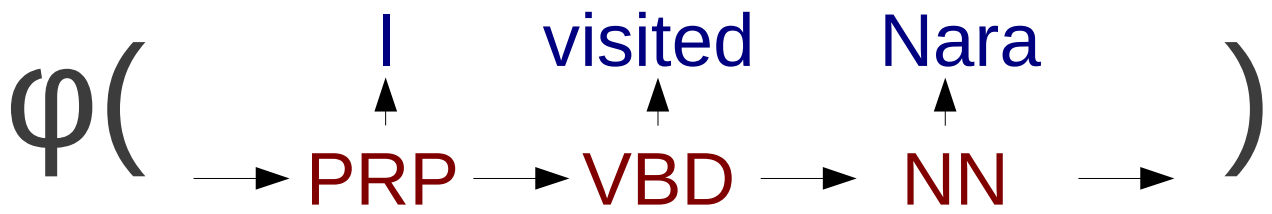
- 負例：正しくないタグ列に対する素性ベクトル

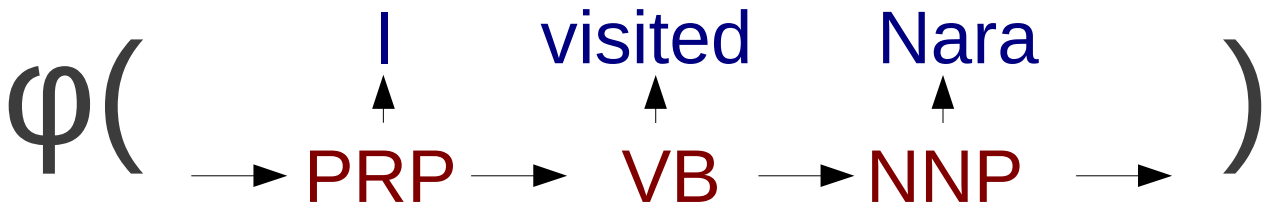


負例の選び方

- 正しくない素性ベクトルがたくさん！







- どれを利用するか？

負例の選び方

- 解決策：スコア最大の正しくない素性ベクトルを利用

$$\hat{Y} = \mathrm{argmax}_Y \sum_i w_i \phi_i(X, Y)$$

- 更新式は：

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \boldsymbol{\phi}(X, Y') - \boldsymbol{\phi}(X, \hat{Y})$$

  - (Y' は正しいタグ列)
- 注：スコア最大のタグ列が正解の場合、変更なし

構造化パーセプトロンアルゴリズム

```
create map w
for / iterations
    for each labeled pair X, Y_prime in the data
        Y_hat = HMM_VITERBI(w, X)
        phi_prime = CREATE_FEATURES(X, Y_prime)
        phi_hat = CREATE_FEATURES(X, Y_hat)
        w += phi_prime - phi_hat
```

HMM の素性計算

- 各遷移、生成に対して素性構築関数を作成

`CREATE_TRANS`(NNP,VBD)

↓

φ[“T,NNP,VBD”] = 1

`CREATE_EMIT`(NNP,Nara)

↓

φ[“E,NNP,Nara”] = 1

φ[“CAPS,NNP”] = 1

構造化パーセプトロンの素性計算

- 単語列の素性を構築する CREATE_FEATURES 関数

```
CREATE_FEATURES(X, Y):
    create map phi
    for i in 0 .. |Y|:
        if i == 0: first_tag = "<s>"
        else:      first_tag = Y[i-1]
        if i == |Y|: next_tag = "</s>"
        else:        next_tag = Y[i]
        phi += CREATE_TRANS(first_tag, next_tag)
    for i in 0 .. |Y|-1:
        phi += CREATE_EMIT(Y[i], X[i])
    return phi
```

素性を使ったビタビアルゴリズム

```
split line into words
l = length(words)
make maps best_score, best_edge
best_score["0 <s>"] = 0   # <s> から開始
best_edge["0 <s>"] = NULL
for i in 0 … l-1:
    for each prev in keys of possible_tags
        for each next in keys of possible_tags
            if best_score["i prev"] and transition["prev next"] exist
                score = best_score["i prev"] +
                        -log P_T(next|prev) + -log P_E(word[i]|next)
                        w*(CREATE_T(prev,next)+CREATE_E(next,word[i]))
                if best_score["i+1 next"] is new or < score
                    best_score["i+1 next"] = score
                    best_edge["i+1 next"] = "i prev"
# </s> に対して同様の処理
```

演習課題

演習課題

- 実装　train-hmm-percep と test-hmm-percep
- テスト
  - 入力：`test/05-{train,test}-input.txt`
  - 出力：`test/05-{train,test}-answer.txt`
- 学習　`data/wiki-en-train.norm_pos`
  実行　`data/wiki-en-test.norm`
- 評価
  `script/gradepos.pl data/wiki-en-test.pos my_answer.pos`
- 比較　通常の生成モデルを用いた HMM と
- チャレンジ
  新しい素性を導入
  平均化、マージン、正則化などの識別学習を利用

Thank You!