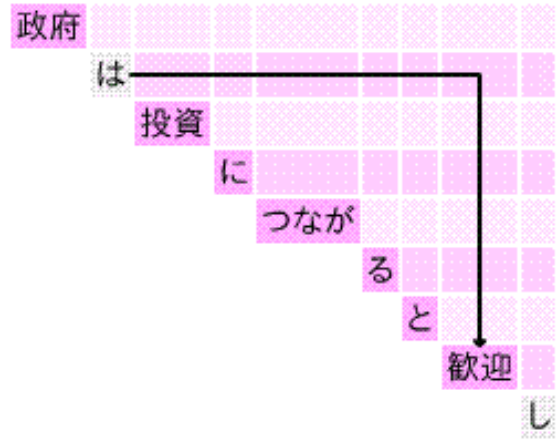
部分的アノテーションから学習可能な係り受け解析器

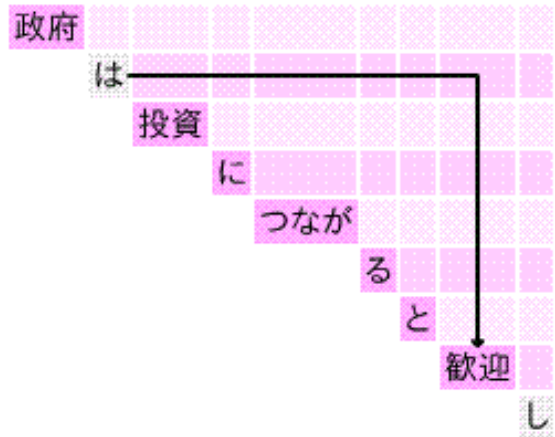
森 信介 [KU]　FLANNERY Daniel[KU]
宮尾 祐介 [NII]　NEUBIG Graham[KU]

京都大学 [KU]　**国立情報学研究所** [NII]

2011 **年** 5 **月** 17 **日**



概要



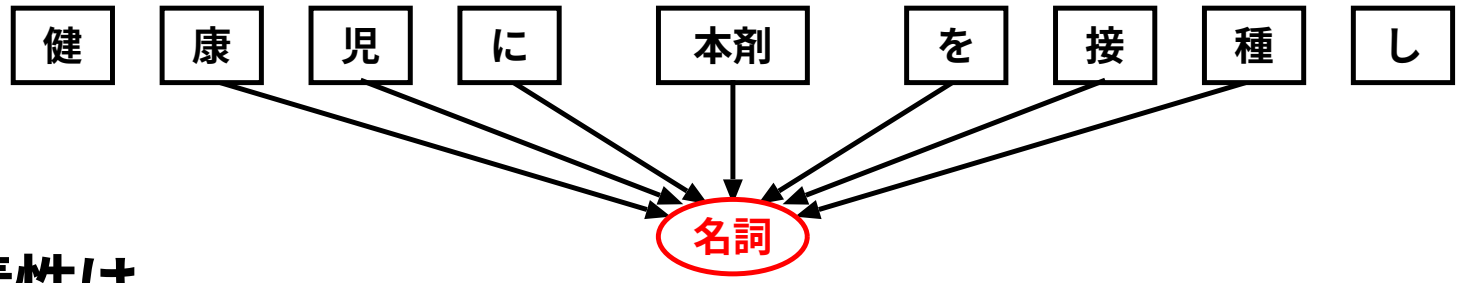
Pointwise NLP
点予測による言語処理

- 推定値を参照しない
- 必要なときに必要なだけの曖昧性解消
- 部分的アノテーションによる迅速分野適応 (Agile Adaptation)

1. 単語分割 [LREC 2010] — KyWS ∈ KyTea
2. 品詞推定 [ACL 2011, NL198] — KyPT ∈ KyTea
3. 読み推定 [LREC 2010] — KyPE ∈ KyTea
4. 係り受け解析 — 名前はまだない

推定値を参照しない

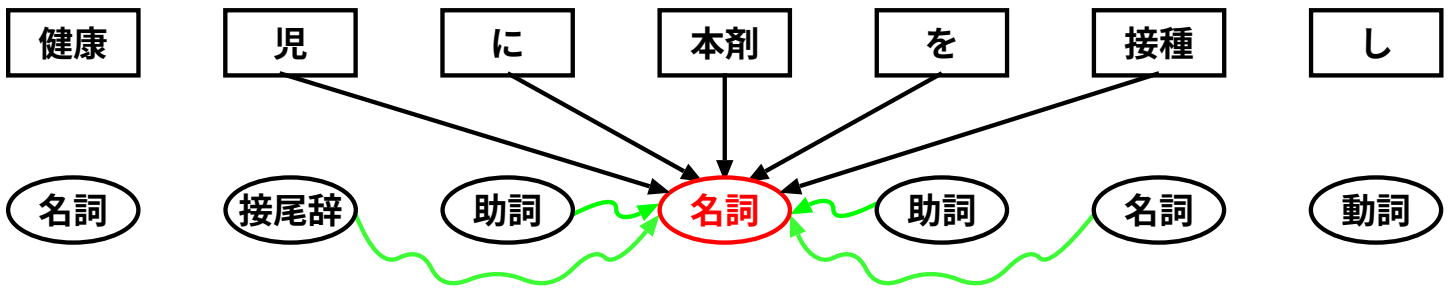
- 点予測による形態素解析



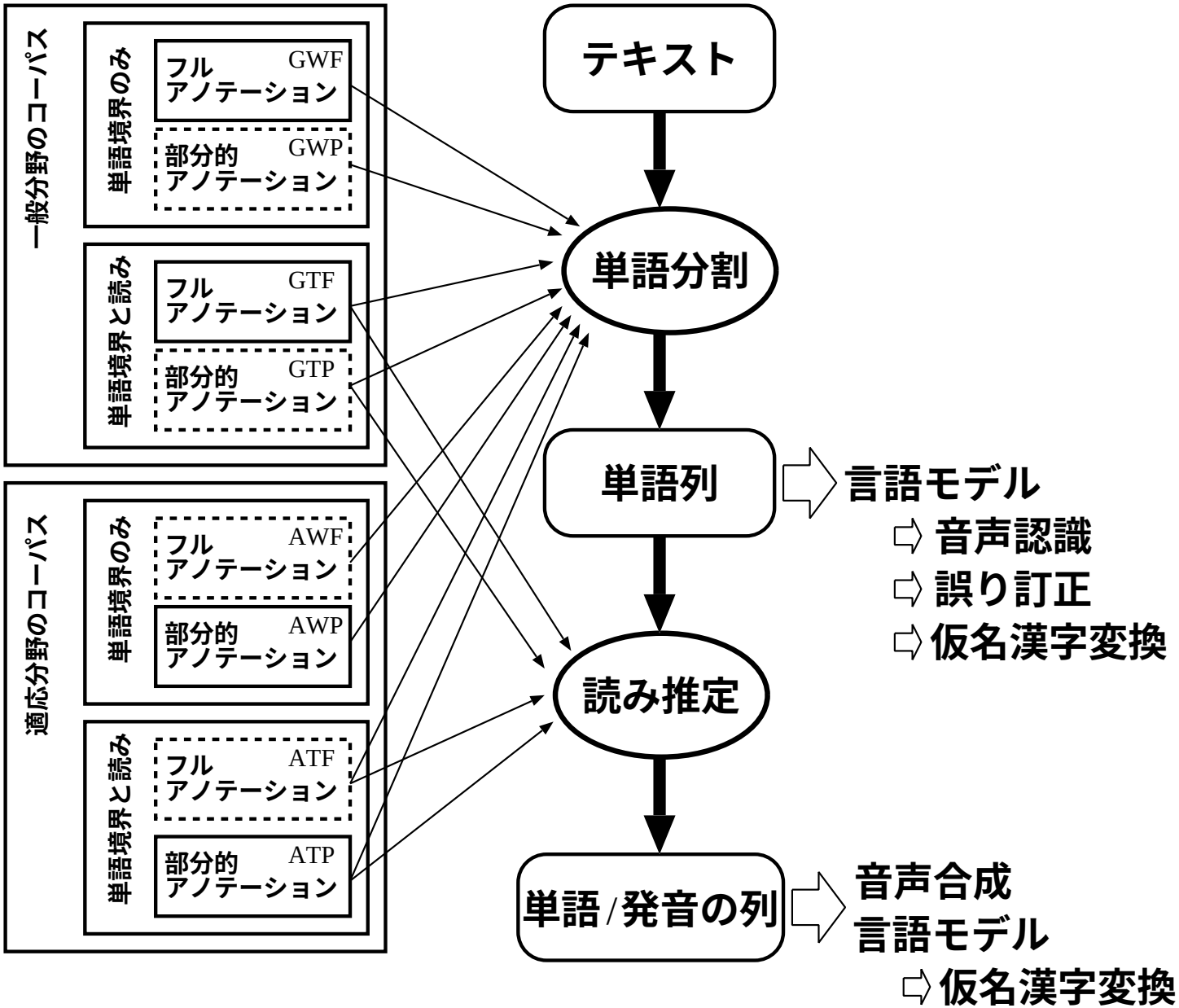
素性は

1. 注目単語
2. その単語境界
3. 前後の文字列

- Cf. 系列に基づく形態素解析 (CRF, n-gram モデル)



必要なときに必要なだけの曖昧性解消



部分的アノテーションによる迅速分野適応

- **部分的アノテーションコーパス**が利用可能
  - **分野特有の表現**のみ情報付与

> 例) ガーゼ等は **本剤/名詞** を吸着する

  - 注目単語の単語境界と品詞のみ

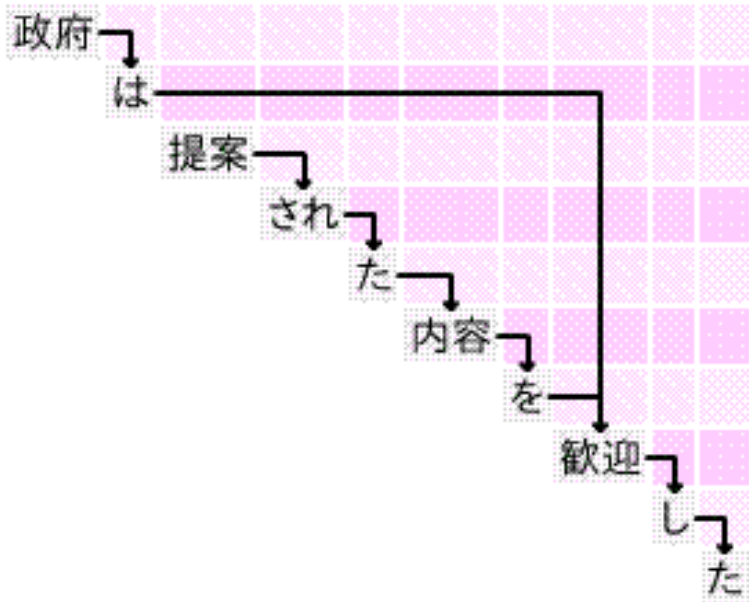
- Cf. **フルアノテーションコーパス**
  - 文のすべての単語境界と品詞

> 例) ガーゼ/名詞 等/接尾辞 は/助詞 本剤/名詞
> を/助詞 吸着/名詞 する/動詞

係り受け解析

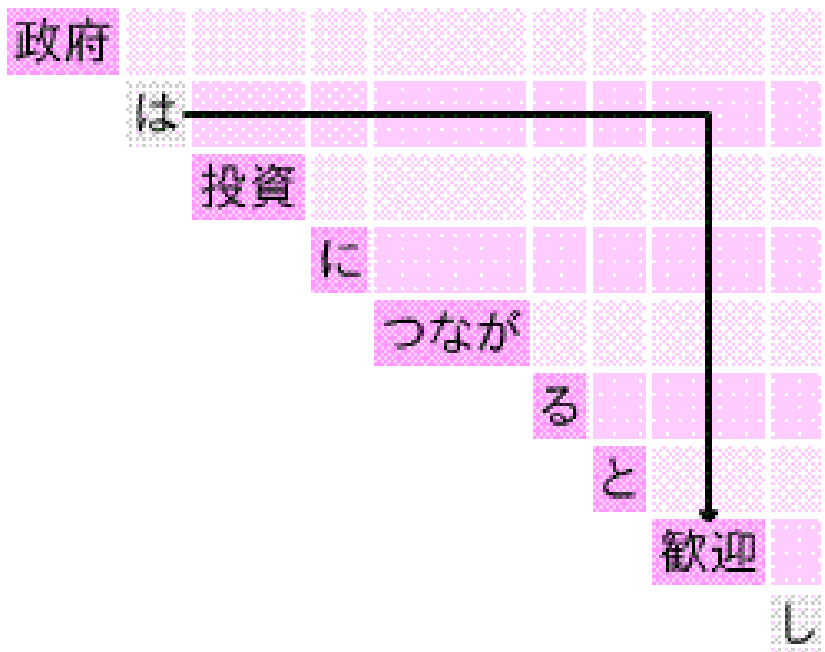
1. **単語をノードとし**
2. **修飾関係をアークとする**
   (いまのところラベルなし)
3. **木構造**

- **MST, Shift reduce, ...**



様々な分野のテキストで高い解析精度の要求

- 学習コーパスと異なる分野では精度が低い
  - Agile Adaptation
  - 迅速分野適応
- 部分的アノテーションからの学習
  - 点予測の方法をどう適用するか



Maximum Spanning Tree

最 大 全 域 木による係り受け解析

[McDonald 他, 2005]

- 入力: $\vec{w} = \langle w_1, w_2, \ldots, w_n \rangle$
- 出力: $\vec{d} = \langle d_1, d_2, \ldots, d_n \rangle$, ($d_i$ は w_i の係り先)

1. ある定義でエッジスコア $\sigma(\langle i, d_i \rangle)$ を計算
2. 以下の最大全域木を探索

$$\hat{\vec{d}} = \operatorname*{argmax}_{\vec{d}} \sum_{d \in \vec{d}} \sigma(\langle i, d_i \rangle)$$

※ 日本語の書き言葉 $\Rightarrow$ 係り先は必ず右にある

点予測によるエッジスコア計算

- 係り先の選択 $\Rightarrow$ n 値分類の問題

$$\sigma(\langle i, d_i \rangle) = p(j|\vec{w}, i), \quad (d_i \in [0, n] \wedge d_i \neq i)$$

- 素性
  1. 係り元と係り先の距離
  2. 係り元/先の表記
  3. 係り元/先の品詞
  4. 係り元/先の前後 3 単語　($\Rightarrow$ 単語分割済みである必要)
     ※周辺は文字列として参照したいが…
- 利点・欠点
  - 利用可能な言語資源の拡大
  - 精度低下?

分野適応

- 一般分野のフルアノテーションコーパス
- 適応分野の部分的アノテーションコーパス

i	1	2	3	4	5	6	7	8	9
w_i	政府	は	投資	に	つなが	る	と	歓迎	し
d_i		8							

1. 「政府-は」$\longrightarrow$「つなが」？ 「歓迎」？
2. 難しい箇所だけ係り受け付与
   - 作業を容易にかつ効率的に

   cf. 読み推定の能動学習 [LREC 2010]

- 係り先が付与されている単語のみ学習

  正例: $w_2 \longrightarrow w_8$, 負例: $w_2 \longrightarrow w_j, (j = 3, 4, \cdots 7, 9)$,

分野適応

- 加算平滑化による確率にしたがって w を選択

$$p = \frac{\alpha + f(w)}{|\mathcal{W}|\alpha + \sum_i f(w_i)} \qquad (\alpha = 0.005)$$

ここで $\mathcal{W}$ は学習コーパスの語彙

- 本発表では部分的アノテーションが使えますという程度
- 能動学習のための評価関数はこれから

評価

- 係り受け解析の精度を比較

1. 既存手法との比較
   - フルアノテーションコーパスでの学習
   - 同一分野のテスト
2. 分野適応
   - 一般分野のフルアノテーションコーパスと
     適応分野の部分的アノテーションコーパスでの学習
   - 適応分野のテスト

コーパスの諸元

ID	出典	用途	文数	単語数	文字数
EHJ-train	辞書の	学習	11,700	145,925	197,941
EHJ-test	例文	テスト	1,300	16,348	22,207
NKN-train	新聞	学習	9,023	263,427	398,570
NKN-test	記事	テスト	1,002	29,038	43,695

- BCCWJ と同じ単語分割基準
- 品詞なし (BCCWJ から構築した KyTea で自動付与)
- EHJ: 一般分野，NKN: 適応分野
- NKN-train は部分的アノテーションのプール

比較手法

- Malt: Nivre ほか (2006) の MaltParser
- MST: McDonald ほか (2005) の MST Parser
- PW: 提案手法
  - 係り受けのエッジスコアの計算は点予測

フルアノテーションコーパス利用

- EHJ-train で学習、EHJ-test でテスト

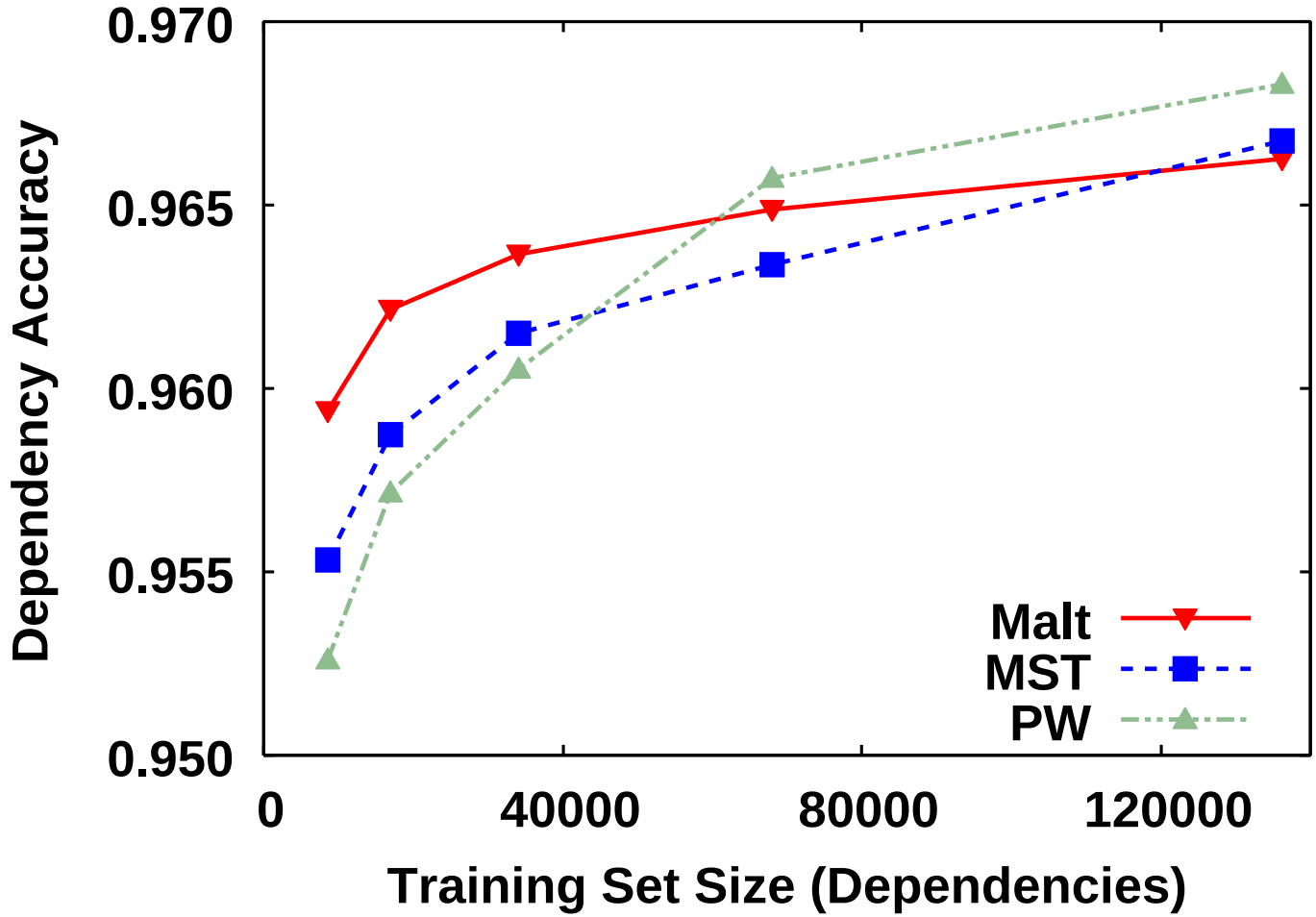
手法	解析精度
Malt	96.63%
MST	96.67%
PW	96.83%

- PW が少し良いがほとんど同じ
- 計算時間 (CPU = 3.33GHz, RAM = 12GB)

手法	学習時間	解析時間
Malt	14[s]	1.3[ms/文]
MST	1901[s]	32.7[ms/文]
PW	125[s]	2.8[ms/文]

- 通常の MST よりかなり速い $\Rightarrow$ 能動学習

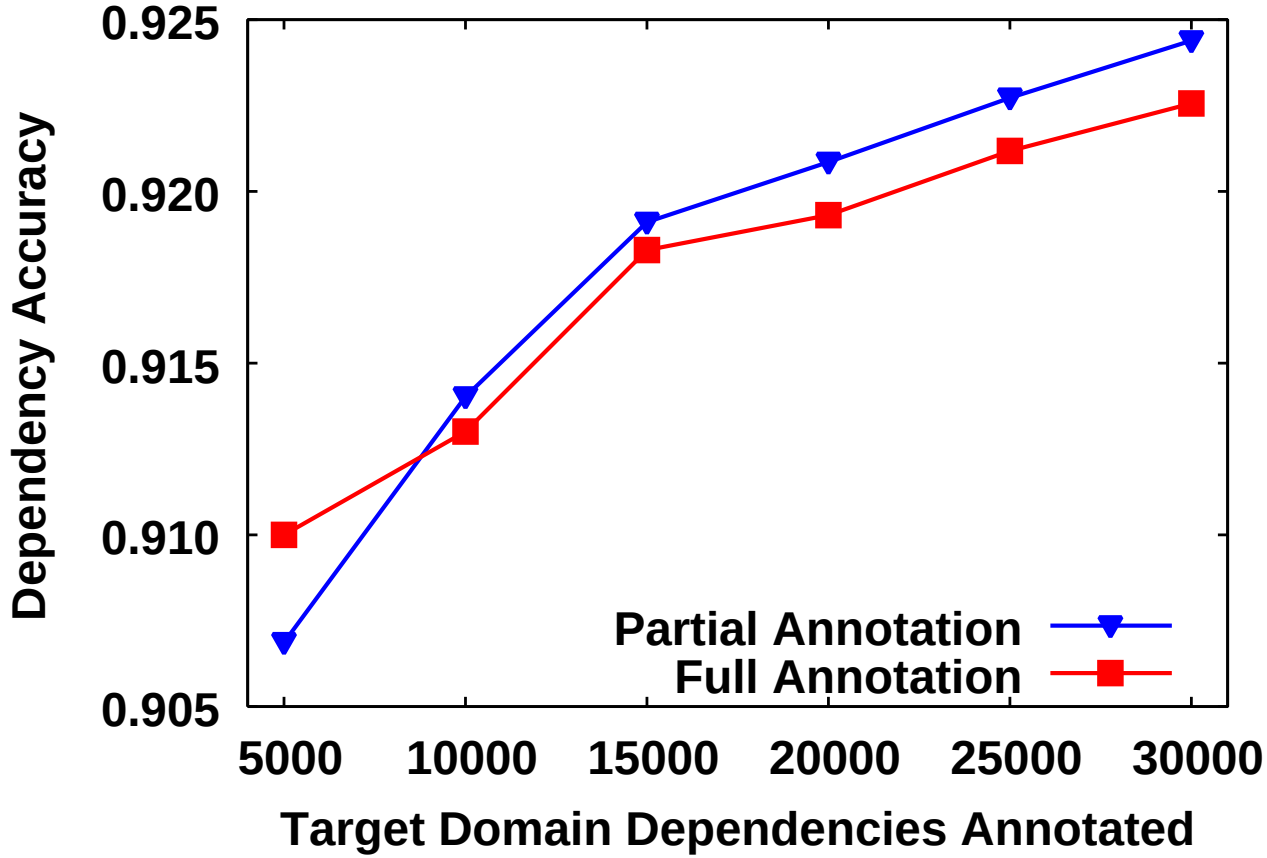
フルアノテーションコーパスサイズ v.s. 精度



- コーパスが大きくなると PW の精度が高くなる (11.47[係り受け/文])

部分的アノテーションによる分野適応

- EHJ-train の全部と NKN-train の一部 (横軸) で学習



- フルアノテーションより速く精度が上がる
- 加算平滑化よりもよい能動学習の基準がある

まとめ

- 単語を単位とする係り受け解析
- 点予測による MST Parser
- 能動学習に使えそうな速度
- フルノテーションの利用では
  通常の MST Parser や Malt Parser と同等の精度
- 部分的アノテーションによる分野適応

To Do List

- **能動学習**
- **様々な言語資源の利用**
  1. **文節ベースのコーパスの利用**
  2. **格フレームなどの部分木の参照**
- **アークラベルの推定**
- **述語項構造の抽出**
- **...**