チャンキングの段階適用による係り受け解析

工藤 拓, 松本 裕治

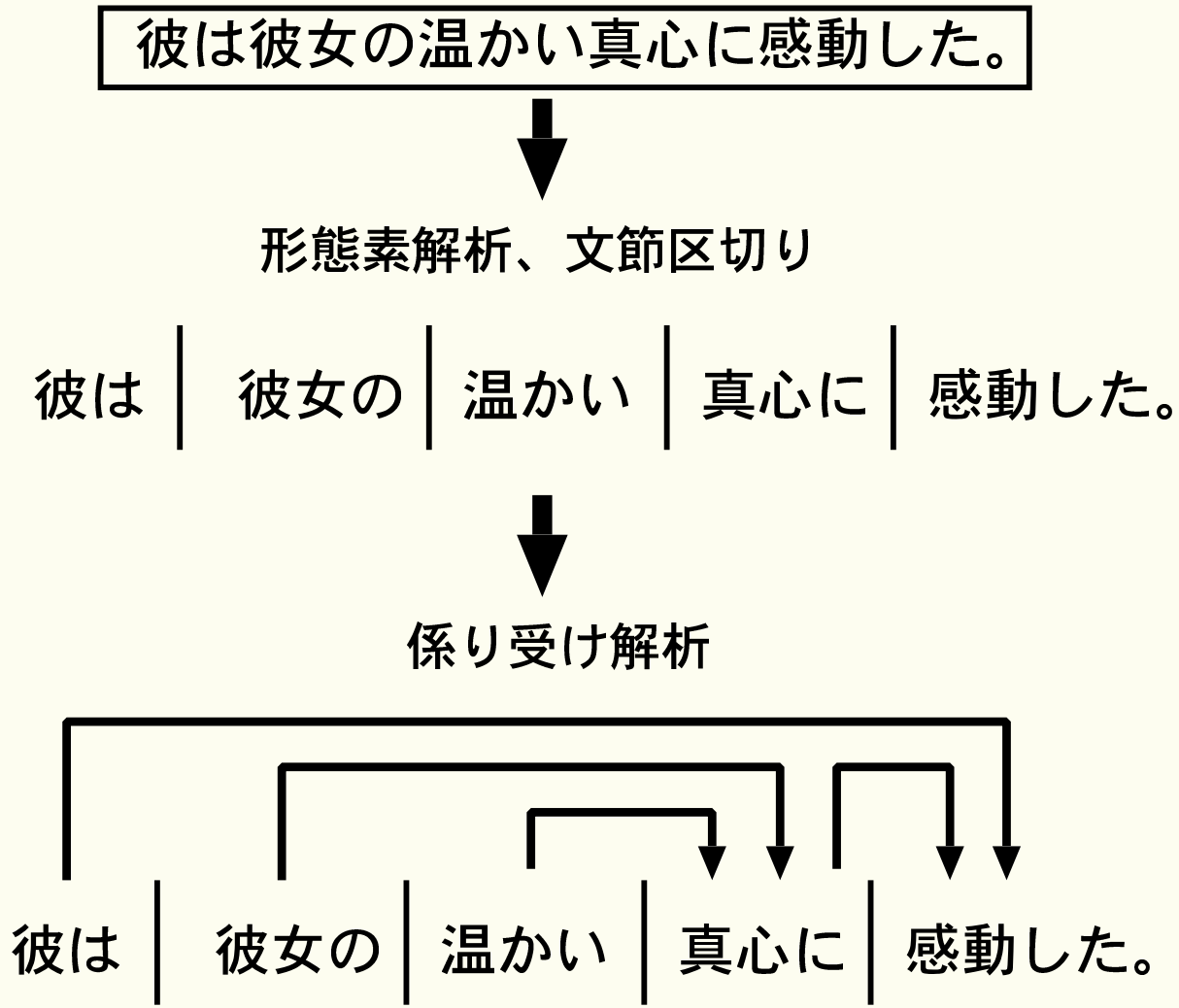
{taku-ku,matsu}@is.aist-nara.ac.jp.

奈良先端科学技術大学院大学 情報科学研究科
自然言語処理学講座

係り受け解析 (1/3)

- 日本語構文解析の基本となる技術
  - **文節** A が **文節** B を修飾する事を A が B に **係る** という
  - 係り受け解析 = 文節の修飾関係の同定
- 2 つの制約
  1. ある**文節** は後方にある一つの文節に係る **(後方参照)**
  2. 係り関係は交差しない **(非交差条件)**

係り受け解析 (2/3)


彼は彼女の温かい真心に感動した。
形態素解析、文節区切り
彼は | 彼女の | 温かい | 真心に | 感動した。
係り受け解析
彼は | 彼女の | 温かい | 真心に | 感動した。

係り受け解析 (3/3)

- 直後の文節に係けるだけ → 約 70%
- 人手によるルールに基づくシステム → 90%
- 統計的手法 (確率モデル, ME, 決定木) → 85 - 88%

係り受け解析 (3/3)

- 直後の文節に係けるだけ → 約 70%
- 人手によるルールに基づくシステム → 90%
- 統計的手法 (確率モデル, ME, 決定木) → 85 - 88%

⇓

- チャンキングの段階適用による高効率なモデル (カスケード モデル) の提案
- 実際の係り関係の推定に SVM を使用

統計的係り受け解析

- 確率モデル (従来法)
- カスケード モデル (提案法)

確率モデル (1/4)

- 係り受け解析に用いられる典型的なモデル
- 二つの文節の係りやすさを示す**確率**を計算
  $M[i,j] = P(Dep(i) = j|\mathbf{f}_{i,j})$ (文節 i が j に係る確率)
- 係り関係はすべて独立と仮定, 文の生成確率は個々の確率の積

$$P(D|S) = \prod_i P(Dep(i) = j|\mathbf{f}_{i,j})$$

- 非交差条件を考慮しながら $P(D|S)$ を最大にする係り関係を探索
  (CYK, 関根 00 等の解析アルゴリズム)

確率モデル (2/4)

Modifiee

彼は　彼女の 温かい 真心に 感動した。

Modifier

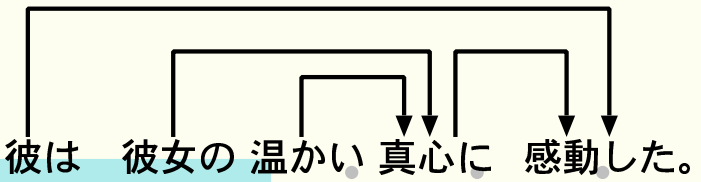
彼は	0.0	0.1	0.2	0.1	0.6
彼女の	0.0	0.0	0.3	0.5	0.2
温かい	0.0	0.0	0.0	0.8	0.2
真心に	0.0	0.0	0.0	0.0	1.0
感動した。	0.0	0.0	0.0	0.0	0.0

$$M[i, j] = P(D(i) = j \mid f_{ij})$$

確率モデル (3/4)

Sekine's backward beam search method (beam width=3)

ID	彼は 1	彼女の 2	温かい 3	真心に 4	感動した。 5	
Cand1				5(1.0)		Prob. 1.0
Cand1			4(0.8)	5(1.0)		Prob. 0.8
Cand2			5(0.2)	5(1.0)		Prob. 0.2
Cand1		4(0.5)	4(0.8)	5(1.0)		Prob. 0.4
Cand2		3(0.3)	4(0.8)	5(1.0)		Prob. 0.24
Cand3		3(0.2)	4(0.8)	5(1.0)		Prob. 0.16
Cand1	5(0.6)	4(0.5)	4(0.8)	5(1.0)		Prob. 0.24 ◀ Best
Cand2	3(0.6)	3(0.3)	4(0.8)	5(1.0)		Prob. 0.14
Cand3	5(0.6)	3(0.2)	4(0.8)	5(1.0)		Prob. 0.08



確率モデル (4/4)

[Kudo, Matsumoto 00]　確率モデル + SVM

- すべての二つの文節の組み合わせのうち, 学習データ中で
  係ったもの → 正例
  係らなかったもの → 負例
- SVM の分類関数 (分離平面からの距離) を Sigmoid 関数に代入し, 擬似確率を求めることで確率モデルの枠組で解析
- 京大コーパスにて 89.09% の精度

確率モデル (4/4)

[Kudo, Matsumoto 00]　確率モデル + SVM

- すべての二つの文節の組み合わせのうち, 学習データ中で
  係ったもの → 正例
  係らなかったもの → 負例
- SVM の分類関数 (分離平面からの距離) を Sigmoid 関数に代入し, 擬似確率を求めることで確率モデルの枠組で解析
- 京大コーパスにて 89.09% の精度
- 全二文節を考慮するために, 非効率 (学習に 2 週間)

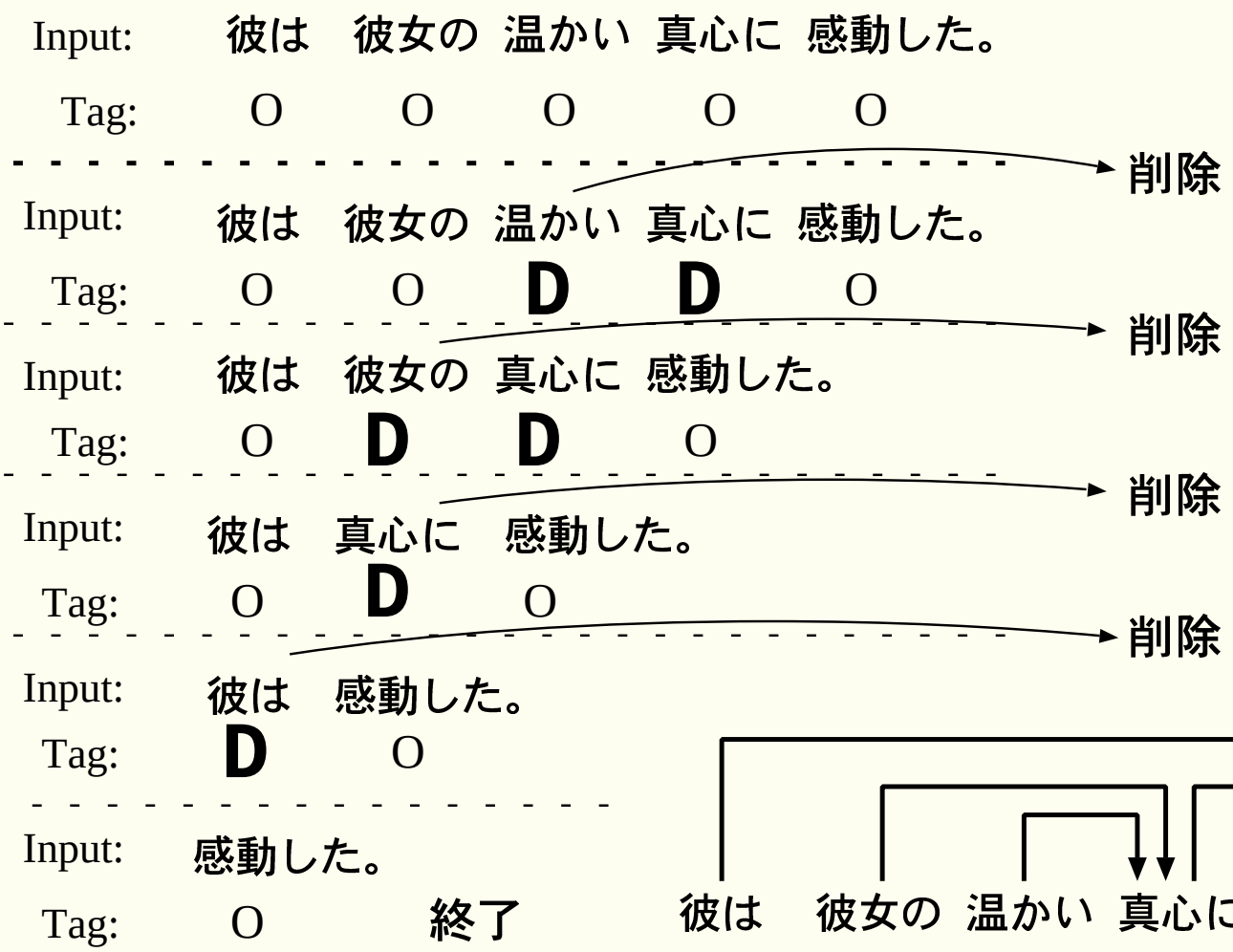
カスケードモデル (1/4)

- 英語の構文解析において適用されている手法,
  これを日本語に適用

カスケードモデル (1/4)

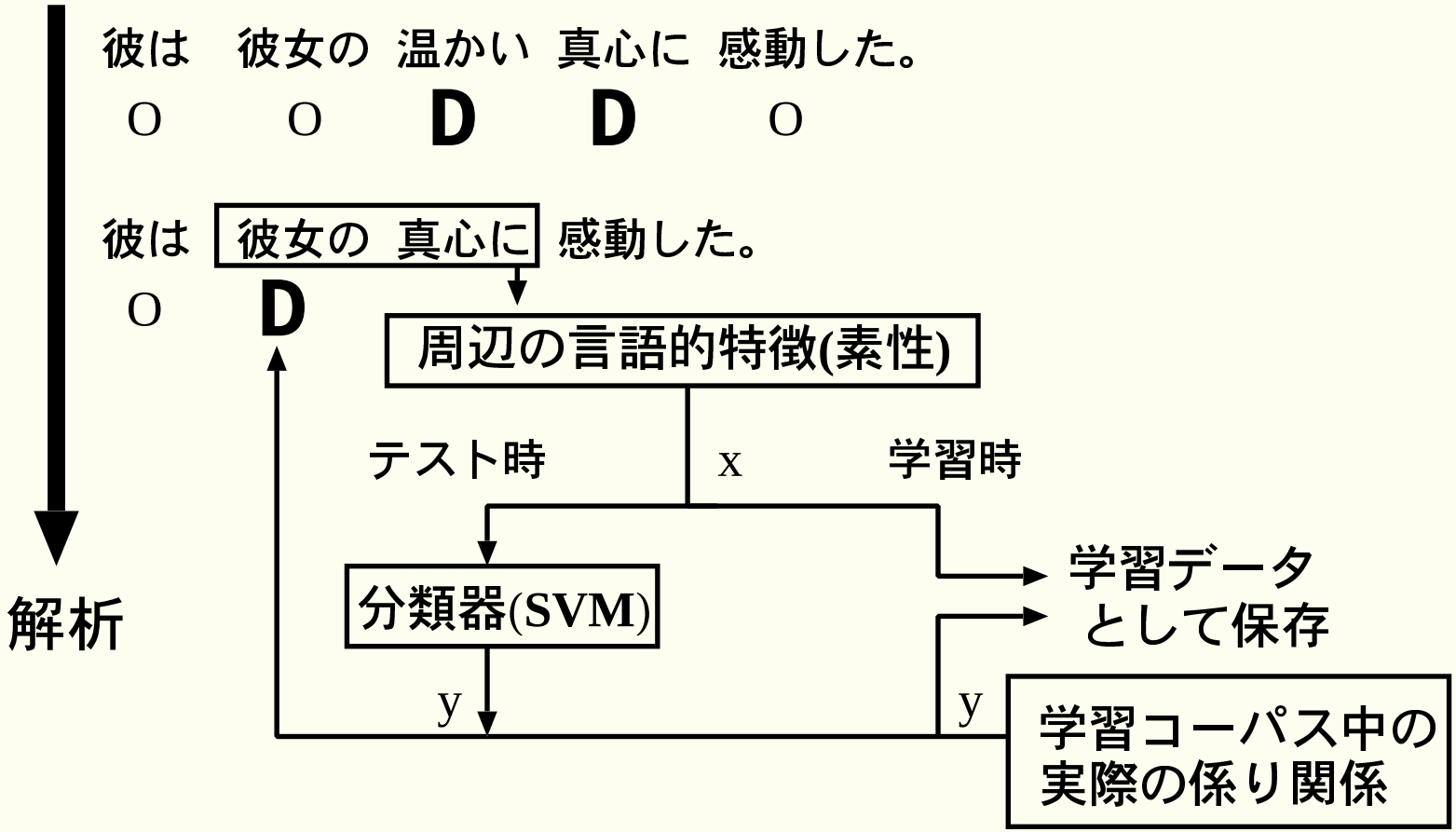
- 英語の構文解析において適用されている手法, これを日本語に適用
- 直後の文節に係るか係らないかという観点で決定的に解析
  1. 全文節に対し, 係り先が未定という意味の O タグを付与
  2. O タグが付与された文節に対し, 直後の文節に係るか推定. 係る場合は D タグを付与.
  3. O タグの直後にある全 D タグとその文節を削除
  4. 残った文節が一つの場合は終了, それ以外は 2. に戻る

カスケードモデル (2/4)


Input: 彼は 彼女の 温かい 真心に 感動した。
Tag: O O O O O
削除
Input: 彼は 彼女の 温かい 真心に 感動した。
Tag: O O D D O
削除
Input: 彼は 彼女の 真心に 感動した。
Tag: O D D O
削除
Input: 彼は 真心に 感動した。
Tag: O D O
削除
Input: 彼は 感動した。
Tag: D O
Input: 感動した。
Tag: O
終了
彼は 彼女の 温かい 真心に 感動した。

カスケードモデル (3/4)

学習とテストが, 解析のプロセスとして同一視



カスケードモデル (4/4)

- アルゴリズムが簡潔, 実装が容易, 高効率

カスケードモデル (4/4)

- アルゴリズムが簡潔, 実装が容易, 高効率
  - 確率: 全二文節が係る/係らない → 学習データ大

カスケードモデル (4/4)

- アルゴリズムが簡潔, 実装が容易, 高効率
  - 確率: 全二文節が係る/係らない → 学習データ大
  - カスケード : 直後に係る/係らない → 学習データ小

カスケードモデル (4/4)

- アルゴリズムが簡潔, 実装が容易, 高効率
  - 確率: 全二文節が係る/係らない → 学習データ大
  - カスケード : 直後に係る/係らない → 学習データ小
- 確率値や尤度は必要ない. 二値分類が行なえる学習アルゴリズムならあらゆるものが適用可能

カスケードモデル (4/4)

- アルゴリズムが簡潔, 実装が容易, 高効率
  - 確率: 全二文節が係る/係らない → 学習データ大
  - カスケード : 直後に係る/係らない → 学習データ小
- 確率値や尤度は必要ない. 二値分類が行なえる学習アルゴリズムならあらゆるものが適用可能
- 非交差条件を自動的に考慮

カスケードモデル (4/4)

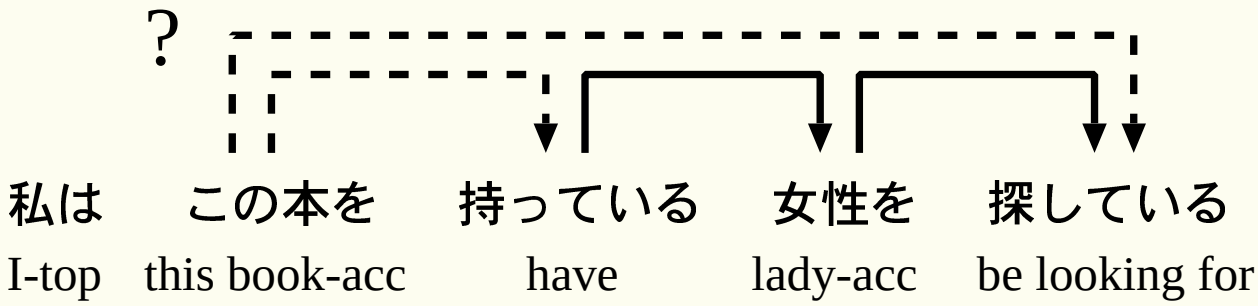
- アルゴリズムが簡潔, 実装が容易, 高効率
  - 確率: 全二文節が係る/係らない → 学習データ大
  - カスケード : 直後に係る/係らない → 学習データ小
- 確率値や尤度は必要ない. 二値分類が行なえる学習アルゴリズムならあらゆるものが適用可能
- 非交差条件を自動的に考慮
- 文頭からの自然な解析

カスケードモデル (4/4)

- アルゴリズムが簡潔, 実装が容易, 高効率
  - 確率: 全二文節が係る/係らない → 学習データ大
  - カスケード : 直後に係る/係らない → 学習データ小
- 確率値や尤度は必要ない. 二値分類が行なえる学習アルゴリズムならあらゆるものが適用可能
- 非交差条件を自動的に考慮
- 文頭からの自然な解析
- 使用するタグを区別するだけで, 係り受けのタイプ (通常の係り受け, 並列, 同格...) を考慮可能

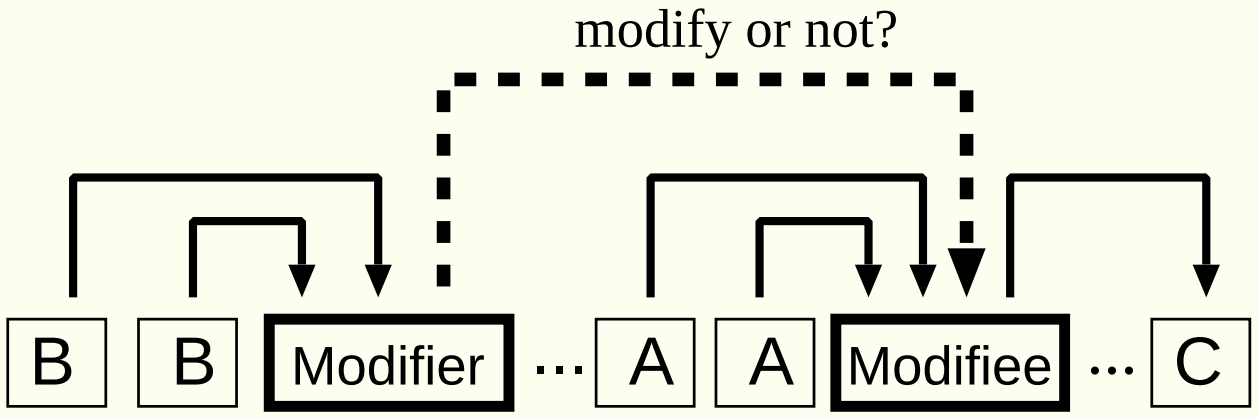
動的素性 と 静的素性 (1/2)

- 静的素性 → 文節区切りが終了した時点で
  決まる素性
- 動的素性 → 係り関係そのもの
  解析を行ないながら動的に追加



動的素性 と 静的素性 (2/2)

1. 着目してる係り先に係る文節 (A)
2. 着目している係り元に係る文節 (B)
3. 着目している係り先が係る文節 (C)



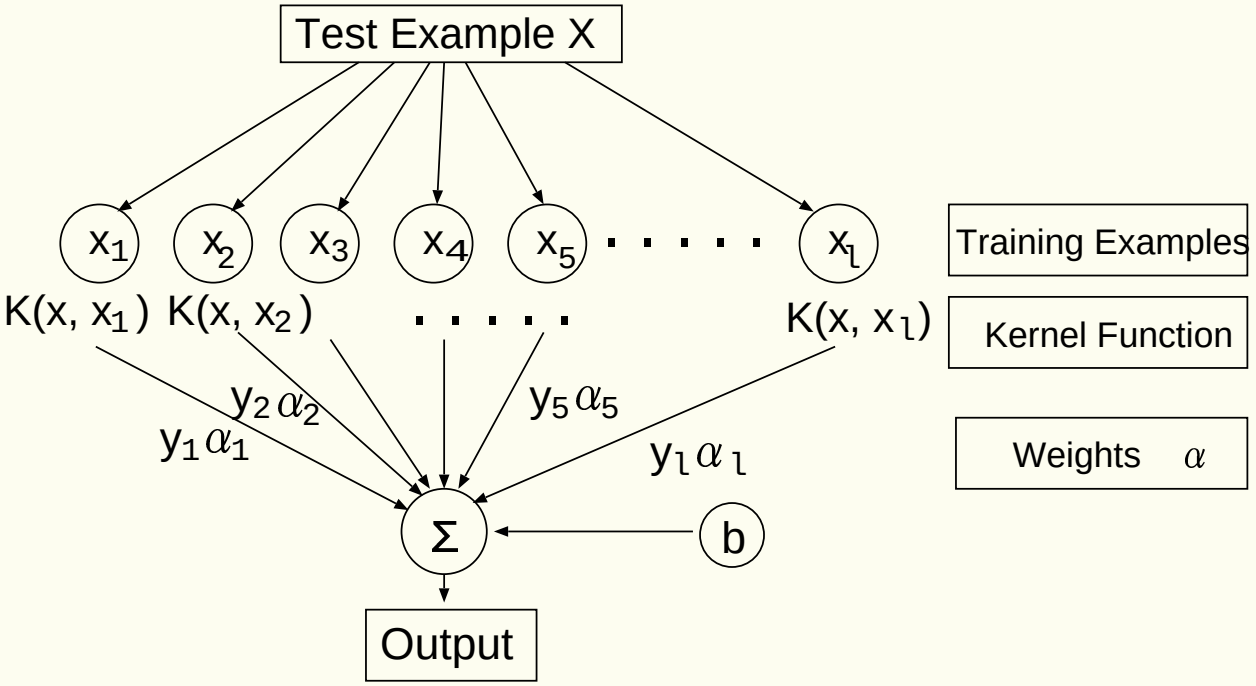
Support Vector Machines
[Vapnik 95,98]

Support Vector Machines (1/2)

学習データ集合: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)$ $\mathbf{x}_i \in \mathbf{R}^n$, $y_i \in \{+1, -1\}$

データ $\mathbf{x}$ から, クラス y への識別関数 $y = f(\mathbf{x}, \theta)$ を導出

$$y = f(\mathbf{x}, \theta) = \mathrm{sgn}\left(\sum_{i=1}^{l} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b\right)$$



Support Vector Machines (2/2)

$$\begin{aligned} &\text{min.} \quad \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &\text{s.t.} \quad y_i \left(\sum_{j=1}^{l} y_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) + b \right) \geq 1 \quad (i = 1, \ldots, l) \end{aligned}$$

- 学習データを誤りなく分類しつつ, 可能な限り最小限のデータで識別関数を表現 ($\alpha_i = 0$ となる事例をできるだけ多く)

Support Vector Machines (2/2)

$$\begin{aligned} &\text{min.} \quad \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &\text{s.t.} \quad y_i \left(\sum_{j=1}^{l} y_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) + b \right) \geq 1 \quad (i = 1, \ldots, l) \end{aligned}$$

- 学習データを誤りなく分類しつつ, 可能な限り最小限のデータで識別関数を表現 ($\alpha_i = 0$ となる事例をできるだけ多く) $\rightarrow$ **オッカムの剃刀**

Support Vector Machines (2/2)

$$\begin{aligned} &\text{min.} \quad \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &\text{s.t.} \quad y_i \left(\sum_{j=1}^{l} y_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) + b \right) \geq 1 \quad (i = 1, \ldots, l) \end{aligned}$$

- 学習データを誤りなく分類しつつ, 可能な限り最小限のデータで識別関数を表現 ($\alpha_i = 0$ となる事例をできるだけ多く) $\rightarrow$ **オッカムの剃刀**
- $\alpha_i > 0$ となる事例 $\mathbf{x}_i$ を **Support Vector** と呼ぶ

Support Vector Machines (2/2)

$$\begin{aligned} &\text{min.} \quad \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &\text{s.t.} \quad y_i \left(\sum_{j=1}^{l} y_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) + b \right) \geq 1 \quad (i = 1, \ldots, l) \end{aligned}$$

- 学習データを誤りなく分類しつつ, 可能な限り最小限のデータで識別関数を表現 ($\alpha_i = 0$ となる事例をできるだけ多く) $\rightarrow$ **オッカムの剃刀**
- $\alpha_i > 0$ となる事例 $\mathbf{x}_i$ を **Support Vector** と呼ぶ
- Kernel 関数の変更により非線型分類が可能

実験, 考察

実験設定

- 京大コーパス Version 2.0
  - 学習データ: 1/1 - 1/8 (7958 文)
  - テストデータ: 1/9 (1246 文)
  - Kernel 関数: 3 次 polynomial (3 個までの素性の組み合わせ, [KM 00] と同一設定)
- 評価方法
  - 係り受け正解率
    (正しく係り先を同定できた割合)
  - 文正解率
    (文全体の係り関係が正しく同定できた割合)

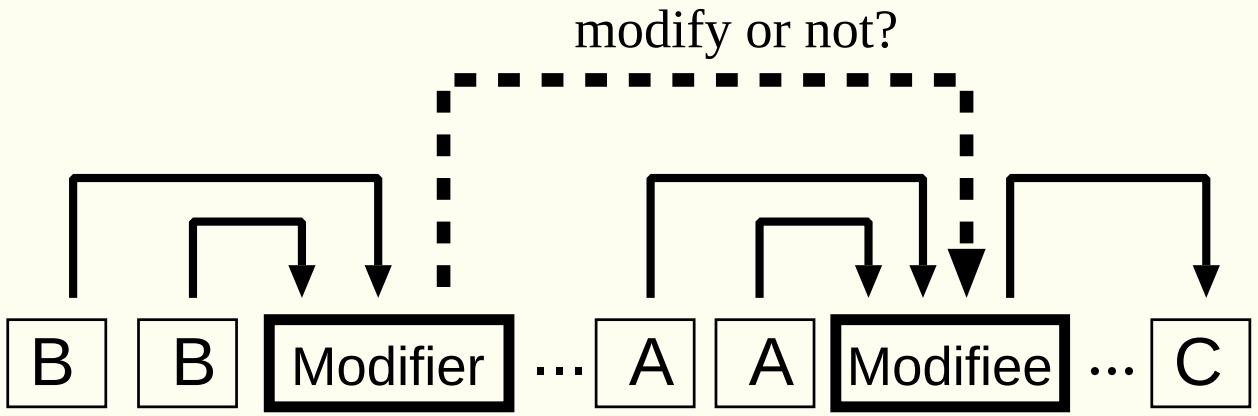
静的素性の設定

前/後文節	主辞見出し, 主辞品詞, 主辞品詞細分類, 主辞活用, 主辞活用形, 語形見出し, 語形品詞, 語形品詞細分類, 語形活用, 語形活用形, 括弧の有無, 句読点の有無, 文節の位置 (文頭, 文末)
文節間	距離 (1,2-5,6 以上), すべての助詞 (は, が, を, に ...), 括弧, 句読点の有無

- 静的素性は [内元 98, 00, KM 00] と ほぼ同一
- すべての語彙を使用 (頻度等によるフィルタリングは行なわない)

動的素性の設定

- (A),(B) → 機能語の部分
- (C) → 主辞の部分

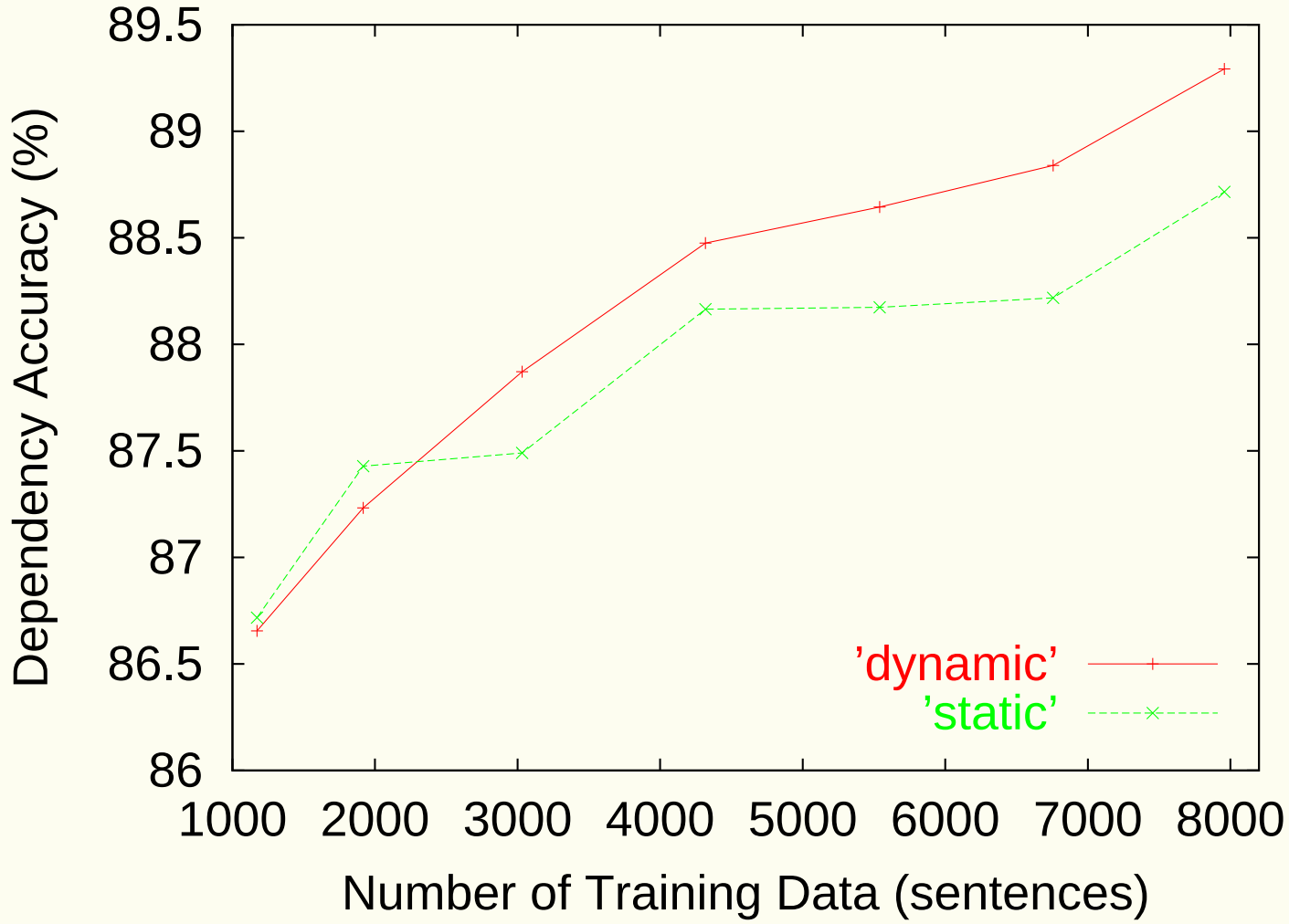


実験結果

確率モデル [KM 00]	係り受け正解率	89.09% (10034/11263)
	文正解率	46.17% (572/1239)
	学習時間	約 2 週間
	解析時間	2.1 秒/文
カスケードモデル (提案法)	係り受け正解率	89.29% (10057/11263)
	文正解率	47.53% (589/1239)
	学習時間	約 10 時間
	解析時間	0.5 秒/文

(学習: AlphaServer 8400, 解析: PentiumIII Linux)

動的素性の効果(1/2)


89.5
89
88.5
88
87.5
87
86.5
86
Dependency Accuracy (%)
1000
2000
3000
4000
5000
6000
7000
8000
Number of Training Data (sentences)
'dynamic'
'static'

動的素性の効果 (2/2)

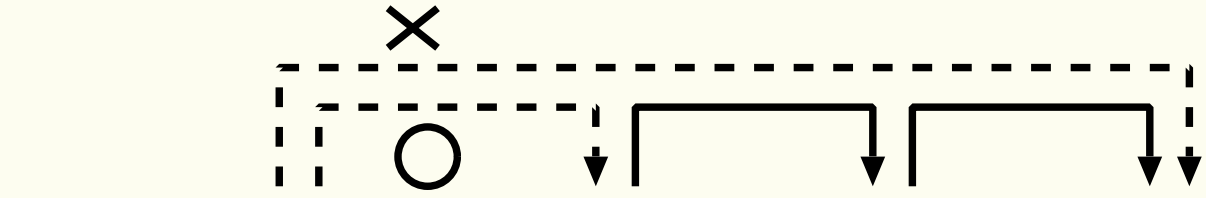
削除した動的素性	精度の増減	
	係り受け正解率	文正解率
A	-0.28%	-0.89%
B	-0.10%	-0.89%
C	-0.28%	-0.56%
AB	-0.33%	-1.21%
AC	-0.55%	-0.97%
BC	-0.54%	-1.61%
ABC	-0.58%	-2.34%

関連研究との比較

内元 00

- Maximal Entropy Model + 確率モデル
- 87.93% (提案手法: 89.29%, 同一学習, テストデータ)
- ME は素性の独立性を仮定.
  内元らは, 素性の組み合わせを人手により発見的に展開　→ 一貫性, 網羅性で問題
- SVM は Kernel 関数の変更のみで, 組み合わせを考慮可能.
  計算量はほとんど変わらない. 係り受け解析に向いている

確率モデル vs カスケードモデル (1/2)



確率モデル: 個々の係り関係は他の関係と独立であると仮定

正例: この本を 持っている

負例: この本を 探している → 極めて例外的な負例

- 多くの文節は後方文脈に依存せず, 直後に係る
- 不必要に多くの例外的事例を学習する可能性
- 真の係り関係を越える関係をすべて負例とするのは安直なのではないか

確率モデル vs カスケードモデル (2/2)

	確率モデル	カスケードモデル
戦略	文全体のコストを重視	直後に係りやすいという heuristics を重視
長所	複数の候補がある場合，それぞれを検出可能	高効率，heuristics に則した自然な解析
欠点	非効率，不必要に例外的な事例を学習	後方にある別の係り先候補を考慮できない

確率モデル vs カスケードモデル (2/2)

	確率モデル	カスケードモデル
戦略	文全体のコストを重視	直後に係りやすいという heuristics を重視
長所	複数の候補がある場合，それぞれを検出可能	高効率，heuristics に則した自然な解析
欠点	非効率，不必要に例外的な事例を学習	後方にある別の係り先候補を考慮できない

これら二つのモデルの融合．複数の候補がある時のみ，それぞれを考慮するモデルの構築

並列構造解析 (1/2)

- 京大コーパスの並列タグの推定
  D,P,O の 3 種類のタグを付与することで実現
- pairwise 法により SVM を多値分類器に拡張
- ルールベースの係り受け解析器 KNP と比較

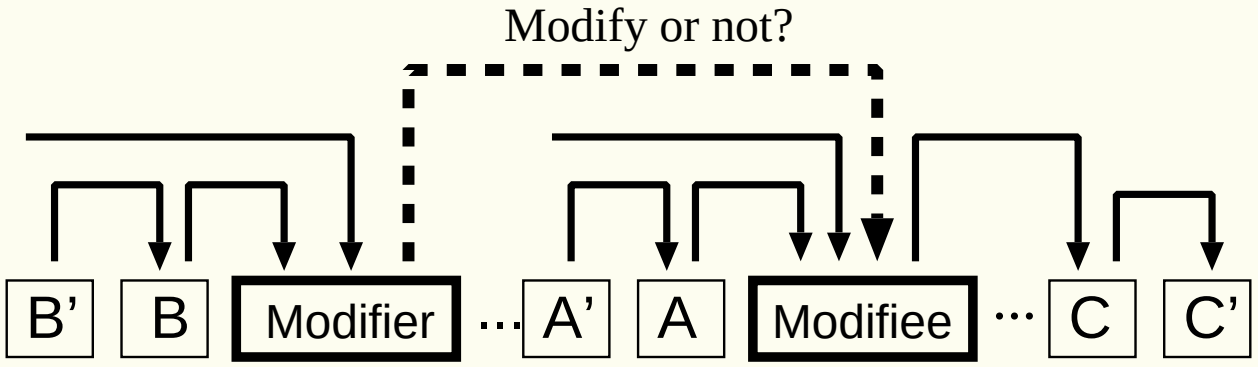
	KNP	提案手法
係り受け正解率	89.68%	89.41%
並列構造正解率	76.32%	65.87%

局所的な関係のみを考慮していることに起因

文全体を考慮した素性を投入する必要性

並列構造解析 (2/2)

- 再帰的な動的素性
- 不必要に素性を追加すると過学習に陥いる可能性
- 精度向上に繋がる最適な素性の選択



後方文脈の考慮

- 内元 00
  係り受け解析を 「手前」「係る」「越える」の三値分類問題にし, それぞれの確率の積を使用
  - 個々の関係は**独立**であると仮定
  - 全係り関係を考慮するために非効率
- 金山 00
  文法により荒い解析をし, 後方の複数の候補のを確率の条件部に入れて推定, **従属性** を考慮

後方文脈の考慮

- 内元 00
  係り受け解析を 「手前」「係る」「越える」の三値分類問題にし, それぞれの確率の積を使用
  - 個々の関係は**独立**であると仮定
  - 全係り関係を考慮するために非効率
- 金山 00
  文法により荒い解析をし, 後方の複数の候補のを確率の条件部に入れて推定, **従属性** を考慮

⇓

金山の手法を採用, 二段階解析

まとめ

- チャンキングの段階適用による新しい係り受け解析手法 (カスケードモデル) の提案
- 従来の確率モデルと比較して高効率
- 7958 文という少量の学習データにもかかわらず 89.29% という高い精度を示す
- 動的素性の考慮により精度がさらに向上