SVM を用いたチャンキングタスクにおける素性の自動選択

工藤 拓, 山田 寛康, 中川 哲治, 松本 裕治

{taku-ku,hiroya-y,tetsu-na,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

チャンキング

- 文 (相当のもの) をある基準で分割し , まとめ上げる (同定)
- まとめ上げた各要素にタグを付ける (分類)

⇓

日本語わかち書き , 英語 tokenization, 形態素解析, 文節切り , baseNP 抽出, Named Entity 抽出, 専門用語抽出...

Chukning → Tagging

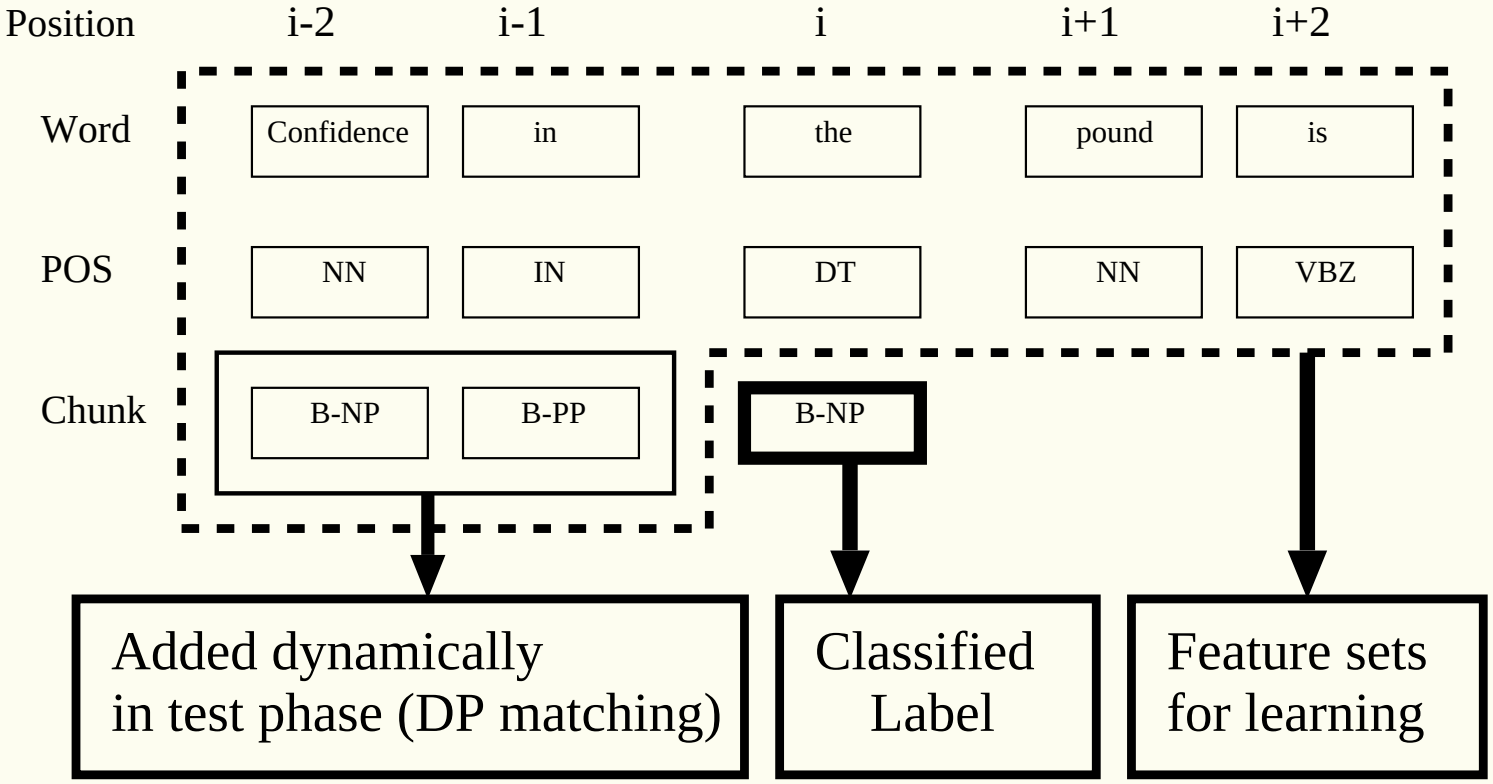
	Base NP Chunking	Chunking
In	O	B-PP
early	B	B-NP
trading	I	I-NP
in	O	B-PP
busy	B	B-NP
Hong	I	I-NP
Kong	I	I-NP
Monday	B	B-NP
,	O	O
gold	B	B-NP

SVM による チャンキング (1/3)

- チャンキングを**タグ付与問題**として扱う
- 従来からあるタグ付与問題 (POS tagging など) の技術が容易に応用可能
- タグの周辺のコンテキストを**素性** とみなし, 現在のタグを精度よく推定するルールを導く処理 → 機械学習の応用
- 二値分類器 の SVM を多値分類器へ拡張 → *pairwise* 法を使用

SVM による チャンキング (2/3)

[工藤 2000]



着目している単語の前後２つの文脈に発見的に固定

SVM による チャンキング (1/3)

- 素性 (文脈長) の影響力の調査
- 汎化誤差を最小にするという観点から最適な素性 (文脈長) を選択する手法を提案

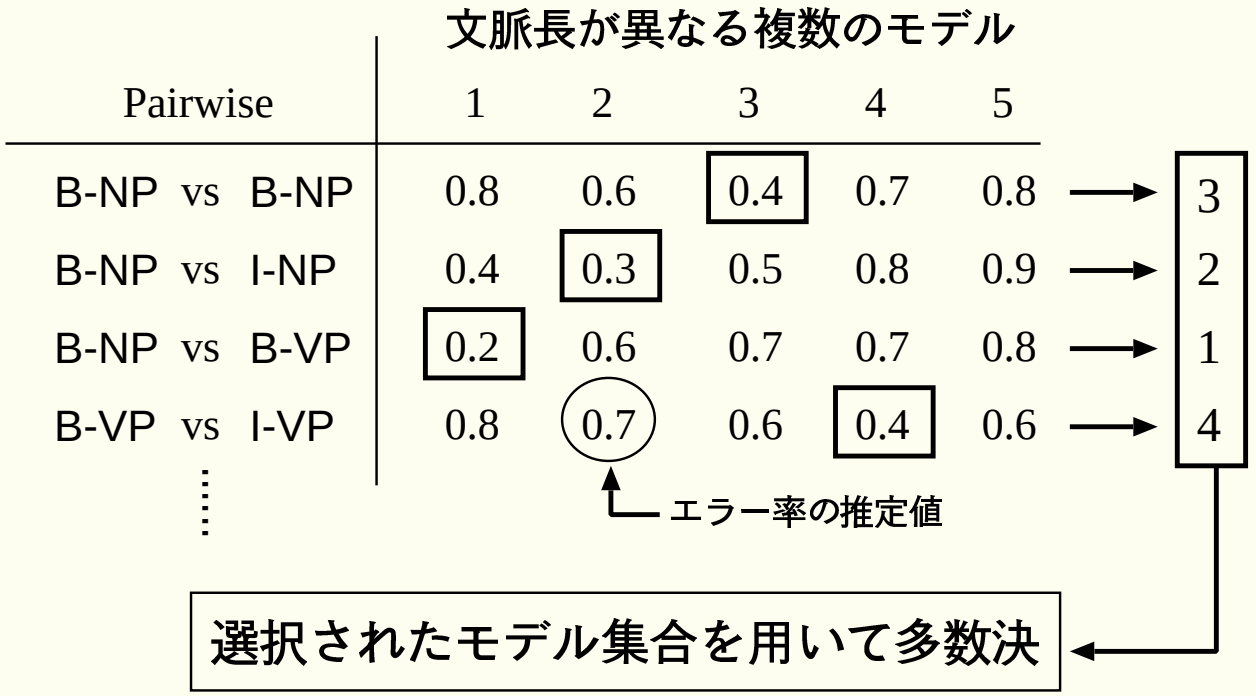
文脈長の影響調査 (Chunking)

文脈長の影響調査 (POS Tagging)

影響調査から分かったこと

- 後方文脈を考慮しないモデルの精度が極めて低い → Beam Search をせずに決定的に解析を行うため.
- 長い文脈を取っても 顕著な精度低下はない → SVMs の持つ素性の次元数 (与えられた素性の数) に依存しない汎化能力の裏付け
- しかし, 最適な文脈長が存在する

(素性)文脈長の自動選択



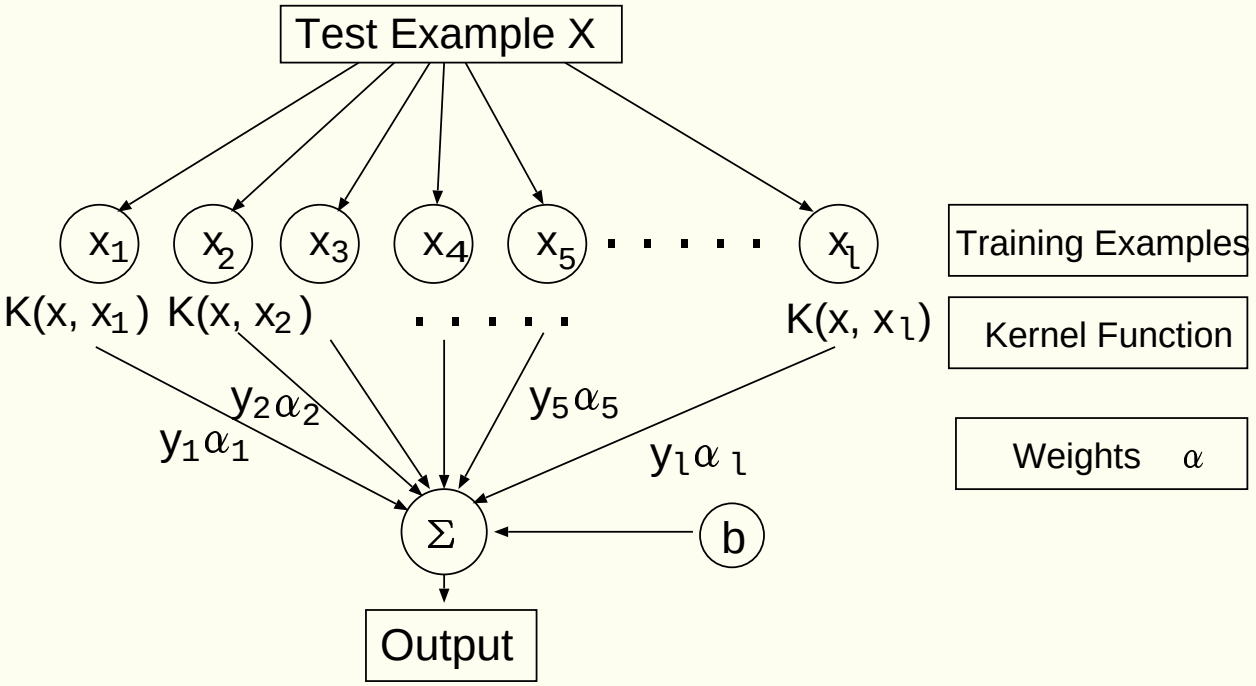
- 基本的なアイデアは単純
- どうやって 与えられた学習データのみから **真のエラー率**を推定するのかが鍵

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)

$$\text{min.} \quad \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)$$

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

- 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

Leave One Out 推定

- l 個の学習データのうち 1 個をとりのぞいてテストデータとし, 残り $l-1$ を使って学習することをすべてのデータについて l 回くりかえす
- 学習アルゴリズムに依存しないモデル選択手法のひとつ

$E_l[f]$ を *Leave-One-Out* によって評価されるエラー率, m を Support Vector の数, l を学習データの数とすると

$$E_l[f] \leq \frac{m}{l}$$

$\xi - \alpha$ 推定

[Joachims 2000] Leave-One-Out のよりタイトな推定方法

$$E_l[f] \leq \frac{Card\{i : 2\alpha_i R^2 \geq 1\}}{l}$$

$$R^2 = \max_{i,j}(K(\mathbf{x}_i, \mathbf{x}_i) - K(\mathbf{x}_i, \mathbf{x}_j))$$

- SVM は, 例外的事例に対し , それ自身を特別視し , 大きな重み α_i を付与し分類を試みる
- α_i が大きいとエラーとしてカウントされやすい

自動選択の結果 (Chunking)

自動選択の結果 (POS Tagging)

考察 (1/2)

- Chunking データセットは, 二つの選択基準によらず, 他のモデルのどれよりも精度が向上.
- POS Tagging のデータセットは *Leave-One-Out* のみが向上. $\xi - \alpha$ は最長文脈長を選んだモデルよりも精度が低く期待どおりの結果ではない

考察 (2/2)

- Chunking に比べ, POS Tagging は個々の pairwise classifier に使われる学習データの量が少ない. (Chunkig は 22 種類のタグ, POS Tagging は, 45 種類のタグ)
- 学習データの数が少ないと, Leave-One-Out 法は汎化能力を過大評価してしまう
- Chunking は学習データが多く, よりタイトな $\xi - \alpha$ が有効に機能

今後の課題

- あらかじめ文脈長の異なる複数のモデルを作成する必要があり効率が悪い
- 本手法は, 文脈長は, 推定すべき**タグ**によって変化する. 本来ならば, 個々の状況 (現在の単語や品詞, 過去に推定したタグ) によって変化するのが自然
- 各状況において 文脈長を adaptive に選択する新たな手法の提案

まとめ

- SVM を用いたチャンクの同定問題における素性 (文脈長) を汎化誤差を最小にするという観点から自動選択する手法を提案
- Leave-One-Out に基づく 二つの推定方法を適用
- 学習データが十分存在する場合は, 提案手法により自動的に文脈長が選択され精度向上が確認された