

# 確率モデルを利用したWeb文書からの見出し抽出

吉田 稔、中川 裕志(東京大学)

## 目的

Webページから見出しを抽出

**Minoru YOSHIDA**

Ph.D. Student  
Department of Information Science, University of Tokyo

Gender: Male  
Nationality: Japanese  
Current Research Interest: Information Extraction  
E-Mail: ymno@is.s.u-tokyo.ac.jp

旧ホームページは [こちら](#)。  
趣味のページは [こちら](#) へ移転中です。

**Curriculum Vitae**

Education and Awards:  
2000 - 2001 Ph.D. course student  
1998 - 2000 Department of Information Science, Faculty of Science, University of Tokyo; Awarded the degree of MSc in Information Science entitled "A Method for Information Extraction from Tables"  
1994 - 1996 Department of Information Science, Faculty of Science, University of Tokyo; Awarded the degree of BSc in Information Science

例  
タイトル(ページ全体を支配)  
ヘッドライン(あるセクションを支配)  
属性(属性値を支配)  
名前(あるオブジェクトの説明部分を支配)  
日付(ある日付の出来事を支配(日記など))  
番号(リストのある要素を支配)

## 応用例

- 属性マイニング: 「名前」と入れたときに「趣味」「性別」等を予測
- 属性サーチ: 検索結果を簡潔に表形式で表示
- 「AのBサーチ」: 「吉田」の「住所」、といった検索が可能に
- Layout Changer: 2つの文書のレイアウトを入れ替える

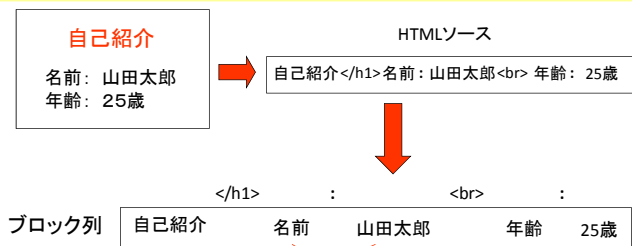
## 既存手法

### 教師あり学習

- ◆SVMでタイトルを抽出(Li et al, 05)  
タイトルだけに特化、WordやPowerPointも対象  
文字の大きさや文書内の位置が重要なfeature
- ◆SVMによる学習(松本 et al, 05)  
自治体Webページを対象

### ルールベース

- ◆発見的ルールによるヘッドライン抽出(Tatsumi et al, 05)  
特定企業Webページを対象  
DOM Treeを対象、HTMLタグのみを扱う
- ◆HTMLタグの繰り返し検出(Mukherjee et al, 03)  
DOMベース(記号は扱えない)
- ◆CFGによる解析(Yoshida and Nakagawa, 07)  
一部教師無し学習も使用  
(「見出し/見出し以外」の2クラスのみ)



- (例)ブロック「年齢」に関する文脈
- ◆左レイアウト: "<br>"
  - ◆右レイアウト: ":"
  - ◆直前ブロック: "山田太郎"
  - ◆文書ID: "3" (とする)

## Future Work

- ◆モデル改善(精度向上)
- ◆多階層見出しへの対応
- ◆速度向上
- ◆英語対応
- ◆クラス数無限化、etc.

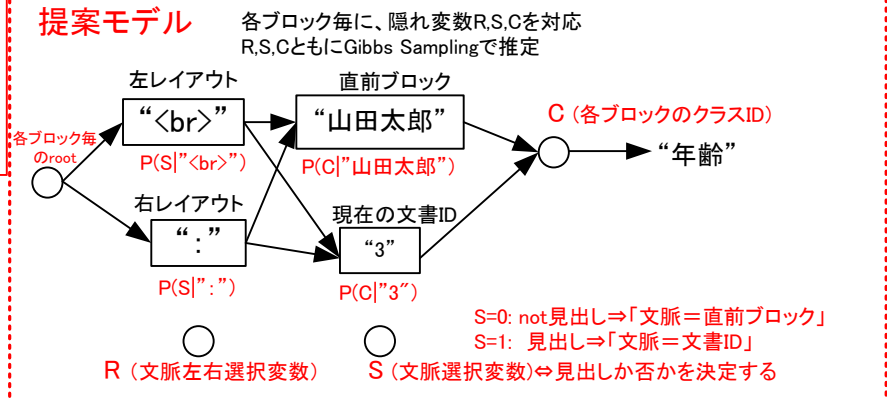
## 提案手法

- ◆レイアウトの多様性
- ◆どんなレイアウトにも対応させたい  
⇒教師無し学習によるアプローチ

## アイデア(仮定と解決策)

- ◆各ブロック文字列は、トピックから生成される。
- ◆文脈に基づくトピック生成(同じ文脈で同じトピック生成⇒高確率)
- ◆見出しに関する仮定に基づき、以下のような文脈を用いる。
  - [仮定1] 見出しは、直前のブロックとの関連が薄い  
{例} "年齢"(見出し)→"25歳"(not見出し): 関連深  
"山田太郎"→"年齢"(見出し): 関連(比較的)浅  
⇒文脈「直前ブロック」
  - [仮定2] 見出しは、同一ページ中に類似のブロックを持つ  
{例} "年齢"(見出し)と"名前"(見出し)が類似  
⇒文脈「文書ID」(≒LDA)
  - [仮定3] 見出しは、見出しに特有のレイアウトを持つ  
{例} ":"が直後に来る、<h2>で囲まれる、等  
⇒文脈「レイアウト(周辺タグ・記号)」
  - [仮定4] 見出しになりやすい語がある  
{例} "住所"や"名前"、"\*月\*日"等  
⇒文脈「ヌル文字列」(文書IDのバックオフで実現)
- ◆各文脈は、ブロック毎に適切なものを選択(≒Pachinko Allocation)
- ◆各文脈には、Hierarchical Pitman-Yor Processによるバックオフスムージングを適用(より短い文脈へのバックオフ)

## 提案モデル



- (1) 左レイアウトか右レイアウトか選択する変数Rを生成
- (2) 周辺レイアウトを基に、見出しか否かを判定する変数Sを生成
- (3) Sを基に文脈を決定し、クラス変数Cを生成
- (4) Cを基に、ブロック(文字列)を生成