# 知的情報処理(11)
# 文書分類: Naïve Bayes と対数線型モデル

櫻井彰人
池山太一

## 文書分類: 例

- ニュース ⇒ *社会、国際、経済、政治、スポーツ、文化・芸能、科学・技術、話題のニュース*
- Medline abstracts に MeSH terms をつける
  - e.g. "Conscious Sedation" [E03.250]
- 商号 ⇒ *産業分野.*
- 学生レポート ⇒ *A,B,C,D.*
- email ⇒ *スパム, その他.*
- 技術員のメール ⇒ *FreeBSD, Linux, Mac, Windows, ..., その他.*
- pdf ファイル ⇒ *研究論文, その他.*
- 文書 ⇒ *重松清(と同一人物)著, その他*
- 映画のレビュー ⇒ *好意的, 批判的, 中立.*
- 研究論文 ⇒ *面白い, 面白くない.*
- ジョーク ⇒ *面白い, 面白くない.*
- ホームページ ⇒ *日本標準産業分類のコード.*

**以下、William W. Cohen (CMU) の Learning to classify text という資料を用いた**

## 文書分類: 例

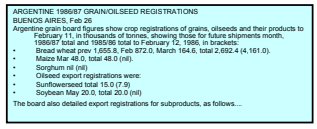
- 頻繁に用いられるベンチマーク: *Reuters-21578* newswire stories
  - 学習: 9603, テスト: 3299, それぞれ 80-100 語, 93 分類クラス

ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS
BUENOS AIRES, Feb 26
Argentine grain board figures show crop registrations of grains, oilseeds and their products to February 11, in thousands of tonnes, showing those for future shipments month, 1986/87 total and 1985/86 total to February 12, 1986, in brackets:
- Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total 2,692.4 (4,161.0).
- Maize Mar 48.0, total 48.0 (nil).
- Sorghum nil (nil)
- Oilseed export registrations were:
- Sunflowerseed total 15.0 (7.9)
- Soybean May 20.0, total 20.0 (nil)

The board also detailed export registrations for subproducts, as follows....

⟶ 分類カテゴリ: **grain**, **wheat** (全部で 93 個. 2値)

## 文書の表現

$f($  $)=y$

**単純かつ有用な**

**?** 分類すべき文書 $x$ の~~最良な~~表現は何か？

## Bag of words 表現

**例**: 単語の出現頻度を用いたテキスト分類の例

ARGENTINE 1986/87 **GRAIN/OILSEED** REGISTRATIONS
BUENOS AIRES, Feb 26
Argentine **grain** board figures show crop registrations of **grains**, **oilseeds** and their products to February 11, in thousands of **tonnes**, showing those for future **shipments** month, 1986/87 **total** and 1985/86 **total** to February 12, 1986, in brackets:
- Bread **wheat** prev 1,655.8, Feb 872.0, March 164.6, **total** 2,692.4 (4,161.0).
- **Maize** Mar 48.0, total 48.0 (nil).
- **Sorghum** nil (nil)
- **Oilseed** export registrations were:
- **Sunflowerseed** total 15.0 (7.9)
- **Soybean** May 20.0, total 20.0 (nil)

The board also detailed export registrations for subproducts, as follows....

⟶ カテゴリ: **grain**, **wheat**

## Bag of words 表現

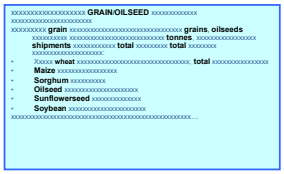
**例**: 単語の出現頻度を用いたテキスト分類の例

xxxxxxxxxxxxxxxxxxx **GRAIN/OILSEED** xxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxx **grain** xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx **grains**, **oilseeds** xxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxx **tonnes**, xxxxxxxxxxxxxxxxx **shipments** xxxxxxxxxxxx **total** xxxxxxxxx **total** xxxxxxxx xxxxxxxxxxxxxxxxxxxxx:
- Xxxx **wheat** xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx, **total** xxxxxxxxxxxxxxxx
- **Maize** xxxxxxxxxxxxxxxxx
- **Sorghum** xxxxxxxxxx
- **Oilseed** xxxxxxxxxxxxxxxxxxxxx
- **Sunflowerseed** xxxxxxxxxxxxxx
- **Soybean** xxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx....

⟶ カテゴリ: **grain**, **wheat**

## Bag of words 表現




| 単語 | 度数 |
|---|---|
| grain(s) | 3 |
| oilseed(s) | 2 |
| total | 3 |
| wheat | 1 |
| maize | 1 |
| soybean | 1 |
| tonnes | 1 |
| ... | ... |

カテゴリ: **grain**, **wheat**

## Bag of words 表現

- 文書 $x$ ⋯ $(w_i, f_i)$ 単語と度数の対
  - $x$ = {(*grain*,3),(*wheat*,1),...,(*the*,6)}
- 各クラス yについて, 確率モデル P(X|Y=y)を求める。
  - P(X={(*grain*,3),...}|Y=*wheat*) = ....
  - P(X={(*grain*,3),...}|Y=*nonWheat*) = ....
- 最適なクラスyを見つける。
  - 確率 P(x|y)に応じて、どのクラスyが最も良いスコアをxに与えるか
  - $f(x) = \text{argmax}_y P(x|y)*P(y)$

## Bayes 規則

$$P(y \mid x) \cdot P(x) = P(x, y) = P(x \mid y) \cdot P(y)$$

$$\Rightarrow \quad P(y \mid x) = \frac{P(x \mid y) \cdot P(y)}{P(x)}$$

$$\Rightarrow \arg\max_y P(y \mid x) = \arg\max_y P(x \mid y) \cdot P(y)$$

## Naive Bayes を用いた文書分類

- P(X|Y) はどう推定するか？
- "bag of words" を生成する最も単純かつ有用な過程:
  - Pr(W|Y) に従って単語1を取り出す
  - 単語2, 3, .... についてこれを繰り返す
  - 各単語は他の単語に対し独立に生成される。
    - これは勿論正しくない。しかし、この結果次のように簡略化される

$$\Pr(w_1,..., w_n \mid Y = y) = \prod_{i=1}^{n} \underbrace{\Pr(w_i \mid Y = y)}$$

**Pr(W|Y) はどう推定するか？**

## Naive Bayes を用いた文書分類

- P(X|Y) はどう推定するか？

$$\Pr(w_1,..., w_n \mid Y = y) = \prod_{i=1}^{n} \Pr(w_i \mid Y = y)$$

*Pr*(*w*|*y*) はデータに従って推定する.

$$\Pr(W = w \mid Y = y) = \frac{\text{count}(W = w \text{ and } Y = y)}{\text{count}(Y = y)}$$

これでは、もし x 中に初めて見る単語 $w_{new}$ があると score は0となってしまう

## Naive Bayes を用いた文書分類

- P(X|Y) はどう推定するか？

$$\Pr(w_1,..., w_n \mid Y = y) = \prod_{i=1}^{n} \Pr(w_i \mid Y = y)$$

... そして **仮に** *m* 個の事例では
*Pr*(*w*|*y*)=*p* であったとしよう

$$\Pr(W = w \mid Y = y) = \frac{\text{count}(W = w \text{ and } Y = y) + mp}{\text{count}(Y = y) + m}$$

用語:
- この Pr(W|Y) は多項分布 ( *multinomial distribution* )となる
- この *m* と *p* は, 多項分布に対するディリクレ事前分布 *Dirichlet prior* で説明できる

## Naive Bayes を用いた文書分類

- P(X|Y) はどう推定するか？

$$\Pr(w_1,\ldots,w_n \mid Y=y) = \prod_{i=1}^{n} \Pr(w_i \mid Y=y)$$

例えば: $m$=1, $p$=0.5

$$\Pr(W=w \mid Y=y) = \frac{\text{count}(W=w \text{ and } Y=y)+0.5}{\text{count}(Y=y)+1}$$

## Naive Bayes を用いた文書分類

- これまでを纏めると:
  - for 各文書 $x_i$（ただし, クラスラベルは $y_i$）
    - for $x_i$ 中の各単語 $w_{ij}$
      - count[$w_{ij}$][$y_i$]++
      - count[$y_i$]++
      - count++
  - 未知文書 $x$=$w_1$...$w_n$ を分類するには, *score*最大の $y$ をとる:

$$score(y, w_1 \ldots w_k) = \log \frac{\text{count}[y]}{\text{count}} + \sum_{i=1}^{n} \log \frac{\text{count}[w_i][y]+0.5}{\text{count}[y]+1}$$

**注:** 文書 $x$ 中に実際に現れた単語についてのみ計算すればよい

## Naïve Bayes for SPAM filtering (Sahami et al, 1998)

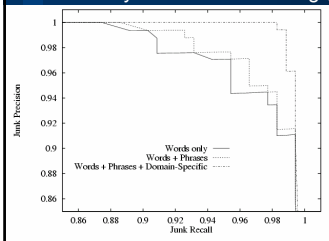



Figure 3: Precision/Recall curves for junk mail using various feature sets.

Used bag of words,
+ special phrases
("FREE!")
+ special features
("from *.edu", …)

**Terms: *precision, recall***

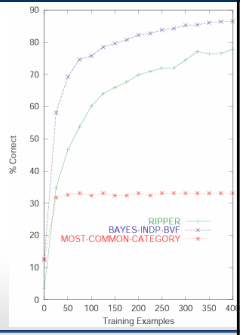
| | Classified Junk | Classified Legitimate | Total |
|---|---|---|---|
| Actually Junk | 36 (92.0% precision) | 9 | 45 |
| Actually Legitimate | 3 | 174 (95.0% precision) | 177 |
| Total | 39 | 183 | 222 |

Table 3: Confusion matrix for real usage scenario.

## Naïve Bayes vs Rules (Provost 1999)

**別の実験**: 規則 (キーワードに基づく詳細なブール式) *vs* Naïve Bayes. 内容に基づく電子メール分類. NB が精度もよく速い.

| | | |
|---|---|---|
| GRACS: | 108 | 5.27% |
| ROBOT: | 175 | 8.53% |
| PSYSCOPE: | 109 | 5.31% |
| NN: | 131 | 6.39% |
| SCHOOL: | 69 | 3.36% |
| CS328: | 691 | 33.69% |
| PERSONAL: | 694 | 33.84% |
| CONNECTIONISTS: | 74 | 3.61% |
| Total examples: | 2051 | 100.00% |




## Twenty News Groups (Joachims 1996)

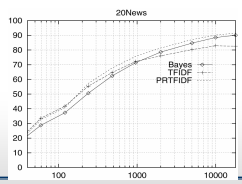
- 各グループ1000の訓練文書
- 新規の文書を、もとのnewsgroupに割振る

| | |
|---|---|
| comp.graphics | misc.forsale |
| comp.os.ms-windows.misc | rec.autos |
| comp.sys.ibm.pc.hardware | rec.motorcycles |
| comp.sys.mac.hardware | rec.sport.baseball |
| comp.windows.x & rec.sport.hockey | rec.sport.hockey |
| alt.atheism | sci.space |
| soc.religion.christian | sci.crypt |
| talk.religion.misc | sci.electronics |
| talk.politics.mideast | sci.med |
| talk.politics.misc | |
| talk.politics.guns | |

T. Joachims. *A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization.* In Proceedings of the 14th International Conference on Machine Learning, Nashville, TN, 1997, pp.143--151.

## Twenty News Groups (Joachims 1996)

- Naive Bayes: 89% 分類正解率
  - 頻出単語上位100 個 (the and of ...) は除去
    - このように文法機能を担う単語や、文書を類別するのに有効でない単語を stop words として除去するのが普通
  - 頻度が3回に満たない単語は除去
  - 残った単語は、約 38,500 語




精度対訓練データ数(1/3はテスト用にとりおいた)

## Log-Linear Model（対数線形モデル）

- Log-Linear Model（対数線形モデル）は自然言語処理をはじめとして多くの分野で使われる確率モデルであり、最大エントロピーモデルやLogistic回帰など多くのモデルがこのモデルに属する。
- Log-Linear Modelの、入力 $x$ に対する出力 $y$ の条件付確率をBayes 規則の単純な式から導出する。

## Naïveさの緩和

$$P(y \mid x) = \frac{1}{Z} P(y) P(x \mid y) \qquad \text{where } Z = P(x) = \sum_y P(x \mid y)$$

$$= \frac{1}{Z} P(y) \prod_{j=1}^{n} P(W_j = w_k \mid y) \qquad \text{文書 } x \text{ に関する } j,k \text{ のみ}$$

$$= \frac{1}{Z} \hat{p}_y \prod_{j,k} \hat{p}_{j,k,y} \qquad \text{文書 } x \text{ に関する } j,k \text{ のみ}$$

**これらの値は、naïve な独立性仮定のもと、推定する**

## Naïveさの緩和

$$P(y \mid x) = \frac{1}{Z} P(y) P(x \mid y)$$

$$= \frac{1}{Z} P(y) \prod_{j=1}^{n} P(W_j = w_k \mid y)$$

$$= \frac{1}{Z} \hat{p}_y \prod_{j,k} \hat{p}_{j,k,y} = \frac{1}{Z} \hat{p}_y \prod_{j,k,Y} \exp\left( (\ln \hat{p}_{j,k,Y}) \cdot \langle W_j = w_k, Y = y \rangle \right)$$

$$= \frac{1}{Z} \lambda_0 \prod_{j,k,Y} \exp\left( \lambda_{j,k,Y} \cdot f_{j,k,Y}(x,y) \right)$$

$$= \frac{1}{Z} \lambda_0 \prod_{i} \exp\left( \lambda_i \cdot f_i(x,y) \right)$$

指示関数 f(x)、重み λ

**指示関数 or 特性関数**
f(x)=1 if condition is true,
f(x)=0 otherwise

**表現の簡略化**

## Naïveさの緩和

- Log-Linear Modelは、入力 $x$ に対する出力 $y$ の条件付確率を次式で表す。

$$P(y \mid x) = \frac{1}{Z} \lambda_0 \prod_i \exp\left( \lambda_i \cdot f_i(x,y) \right) = \frac{1}{Z} \lambda_0 \exp\left( \sum_i \lambda_i \cdot f_i(x,y) \right) \quad \cdots (1)$$

f(x,y)=1 if condition is true,
f(x,y)=0 otherwise

  - $f_i(x,y)$は$(x,y)$のある特徴ベクトル（$j$番目に単語$k$, クラスY）
  - $\lambda$ は各素性に対する重みを並べたモデルパラメータ
    $\lambda$ は訓練事例 $\{(x_1,y_1),\cdots,(x_n,y_n)\}$ を用いた最尤法によって推定

$$\text{maximize} \prod_i P(y_i \mid x_i) \Leftrightarrow \text{maximize} \sum_i \log P(y_i \mid x_i)$$

$$\lambda^* = \arg\max_\lambda \sum_{i=1}^{n} \log P(y_i \mid x_i) \quad \cdots (2)$$

## Naïveさの緩和

$$\lambda^* = \arg\max_\lambda \sum_{i=1}^{n} \log P(y_i \mid x_i) \quad \cdots (2)$$

- 最尤推定は訓練事例数がパラメータ数に対して少ないと過学習を起こす。（特に自然言語処理において素性の種類数は非常に大きい）
  そのため、$\lambda$ に対する正則化項 $r(\lambda)$ を加えた推定を用いる。 スパースなデータも多い

$$\lambda^* = \arg\max_\lambda \sum_{i=1}^{n} \log P(y_i \mid x_i) - C \cdot r(\lambda) \quad \cdots (3)$$

- $C > 0$ はパラメータ。$C$ が小さい場合より訓練データにフィットした学習を行い、$C$ が大きい場合は正則化の影響が強くなる。

ただし、$r(\lambda) = \sum_{i=1}^{m} |\lambda_i|$ または $r(\lambda) = \sum_{i=1}^{m} \lambda_i^2$

## Naïveさの緩和

- これまでを纏めると:
  - $x,y$ 対のありそうな特徴 $f_i(x,y)$ を定める（たくさん）
  - 仮定: $\Pr(y \mid x) = \frac{1}{Z} \lambda_0 \prod_i \exp\left( \lambda_i \cdot f_i(x,y) \right)$
  - 学習: 右式を最大化する $\lambda$'s を求める $\sum_i \log P(y_i \mid x_i)$
    - 最急降下法も ok
      - 最近 (Malouf, CoNLL 2001) Newton法に対する、あるヒューリスティックな近似が、収束を非常に高速化することがわかった
    - 特徴のスパースさには注意
      - ほとんどの特徴は 0
    - 過学習を避けるために: maximize $\sum_i \log P(y_i \mid x_i) - C \cdot r(\lambda)$

## Naïveさの緩和

| Dataset | Method | KL Div. | Acc | Iters | Evals | Time |
|---|---|---|---|---|---|---|
| rules | gis | $5.124\times10^{-2}$ | 47.00 | 1186 | 1187 | 16.68 |
| | iis | $5.079\times10^{-2}$ | 43.82 | 917 | 918 | 31.36 |
| | steepest ascent | $5.065\times10^{-2}$ | 44.88 | 224 | 350 | 4.80 |
| | conjugate gradient (fr) | $5.007\times10^{-2}$ | 44.17 | 66 | 181 | 2.57 |
| | conjugate gradient (prp) | $5.013\times10^{-2}$ | 46.29 | 59 | 142 | 1.93 |
| | limited memory variable metric | $5.007\times10^{-2}$ | 44.52 | 72 | 81 | 1.13 |
| summary | gis | $1.857\times10^{-3}$ | 96.10 | 1424 | 1425 | 107.05 |
| | iis | $1.081\times10^{-3}$ | 96.10 | 593 | 594 | 188.54 |
| | steepest ascent | $2.489\times10^{-3}$ | 96.33 | 1094 | 3321 | 190.22 |
| | conjugate gradient (fr) | $9.053\times10^{-5}$ | 95.87 | 157 | 849 | 49.48 |
| | conjugate gradient (prp) | $3.297\times10^{-4}$ | 96.10 | 112 | 537 | 31.66 |
| | limited memory variable metric | $5.598\times10^{-5}$ | 95.54 | 63 | 69 | 8.52 |
| shallow | gis | $3.314\times10^{-2}$ | 14.19 | 3494 | 3495 | 21223.86 |
| | iis | $3.238\times10^{-2}$ | 5.42 | 3264 | 3265 | 66855.92 |
| | steepest ascent | $7.303\times10^{-2}$ | 26.74 | 3677 | 14527 | 85062.53 |
| | conjugate gradient (fr) | $2.585\times10^{-2}$ | 24.72 | 1157 | 6823 | 39038.31 |
| | conjugate gradient (prp) | $3.534\times10^{-2}$ | 24.72 | 536 | 2813 | 16251.12 |
| | limited memory variable metric | $3.024\times10^{-2}$ | 23.82 | 403 | 421 | 2420.30 |

Table 2: Results of comparison.

## Naïveさの緩和

| | Naive Bayes | Lin Reg | Mod Least Squares | Logistic Reg | SVM | Mod SVM |
|---|---|---|---|---|---|---|
| precision | 77.0 | 87.1 | 89.2 | 88.0 | 89.2 | 89.4 |
| recall | 76.9 | 84.9 | 85.3 | 84.9 | 84.0 | 83.7 |
| $F_1$ | 77.0 | 86.0 | 87.2 | 86.4 | 86.5 | 86.5 |
| BEP | 75.8 | 86.3 | 86.9 | 86.9 | 86.5 | 86.7 |

Table 2: Binary classification performance on Reuters (all 118 classes)

| | Naive Bayes | Logistic Reg | SVM | Mod SVM |
|---|---|---|---|---|
| earn | 96.6 | **98.4** | 98.1 | 98.1 |
| acq | 91.7 | 95.2 | 95.3 | 94.5 |
| money-fx | 70.0 | 75.2 | 74.4 | 74.5 |
| grain | 76.6 | 88.4 | 89.6 | 90.6 |
| crude | 84.1 | 85.9 | 84.8 | 84.0 |
| trade | 52.3 | 72.9 | 73.4 | 74.8 |
| interest | 68.2 | **78.1** | 75.9 | 74.7 |
| ship | 76.4 | 81.9 | 82.4 | **83.8** |
| wheat | 58.1 | 88.2 | 88.9 | **89.6** |
| corn | 52.4 | 88.7 | 86.2 | 86.7 |

**From Zhang & Oles, 2001 – F1 values**

## NLPの研究紹介

池山太一

## 研究紹介①

・高性能計算環境を用いたWebからの大規模格フレーム構築
**Case Frame Compilation**
**from the Web using High-Performance Computing**
河原大輔・黒橋禎夫、情報処理学会、自然言語処理研究会

– Webから日本語文を収集し、コーパスを作成
– 上記Webコーパスから、格フレームを自動構築

• 格フレーム　ex)「積む」
{従業員、運転手、・・・}が{車、ﾄﾗｯｸ、・・・}に{荷物、物資、・・・}を 積む

## 高性能計算環境を用いたWebからの大規模格フレーム構築

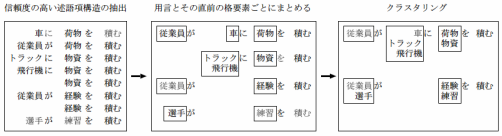



図 3: 格フレーム構築の概要

表 3: 構築した格フレームの例

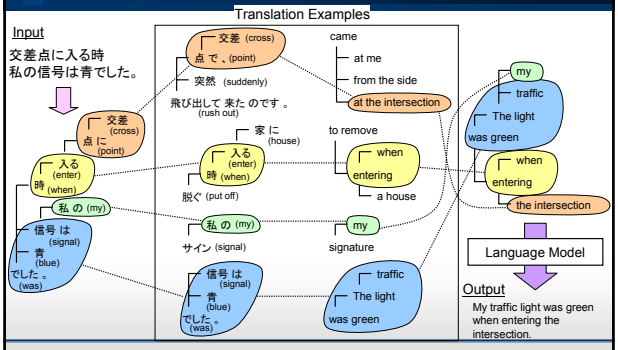
泳ぐ { イルカ, 生, 魚, ..} が { 海, 水中, 海中, ..} を { クロール, 平泳ぎ, バタフライ, ..} で
寝そべる { 人, 男 } が { ビーチ, 砂浜, 浜辺, ..} に
磨く { 私, 男性, 人, ..} が { ブラシ, 所, トイレ, ..} で { 歯, 奥歯, 前歯 } を
⇔ 新聞: { 人, イヌ, 園児, ..} が { 歯 } を 磨く

## 研究紹介②

・Example-based Machine Translation Pursuing Fully Structural NLP
黒橋禎夫・中澤敏明他、2005
・Example-based Machine Translation based on Deeper NLP

– 日本語と英語のペアの例文を用いて、機械翻訳を行う
• 構文情報をもちいている

## Example-based Machine Translation Pursuing Fully Structural NLP




## 簡単な実験

池山太一

## Yahoo!ニュースの見出しに構文解析を適用

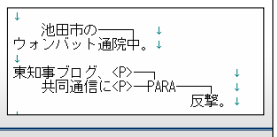
- Yahoo!ニュース　トピックス　の見出しに対して構文解析を行ってみた
  - 国内、地域、経済、海外、エンターテインメント、スポーツ、サイエンス、コンピュータ
    - 各項目7つのニュース

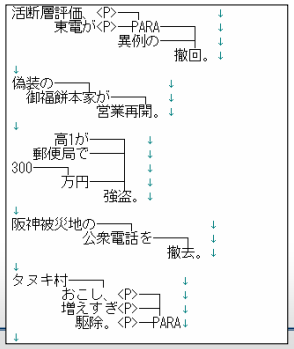
## Yahoo!ニュース　トピックス

- Yahoo! ニューストピック

## 結果＜地域＞

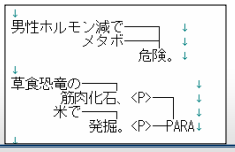
＜地域＞
- 活断層評価、東電が異例の撤回。
- 偽装の御福餅本家が営業再開。
- 高1が郵便局で300万円強盗。
- 阪神被災地の公衆電話を撤去。
- タヌキ村おこし、増えすぎ駆除。
- 池田市のウォンバット通院中。
- 東知事ブログ、共同通信に反撃。

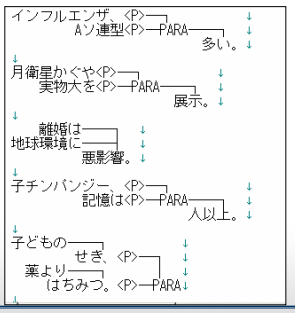



## 結果＜サイエンス＞

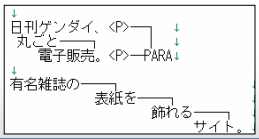
＜サイエンス＞
- インフルエンザ、Aソ連型多い。
- 月衛星かぐや実物大を展示。
- 離婚は地球環境に悪影響。
- 子チンパンジー、記憶は人以上。
- 子どもの せき、薬より はちみつ。
- 男性ホルモン減でメタボ危険。
- 草食恐竜の筋肉化石、米で発掘。

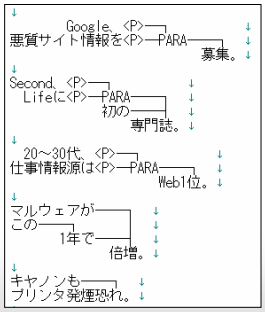


## 結果＜コンピュータ＞

＜コンピュータ＞
- Google、悪質サイト情報を募集。
- Second、Lifeに初の専門誌。
- 20～30代、仕事情報源はWeb1位。
- マルウェアがこの1年で倍増。
- キヤノンもプリンタ発煙恐れ。
- 日刊ゲンダイ、丸ごと電子販売。
- 有名雑誌の表紙を飾れるサイト。



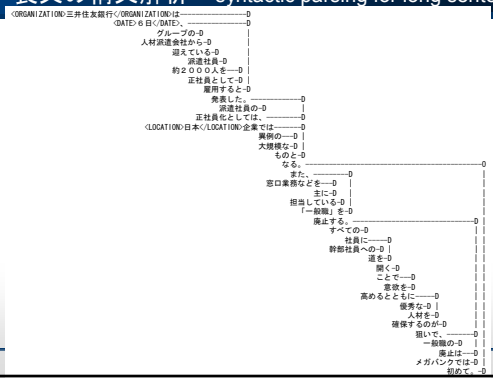

## 長文の構文解析

- ＜三井住友銀行＞派遣社員2000人を正社員化へ

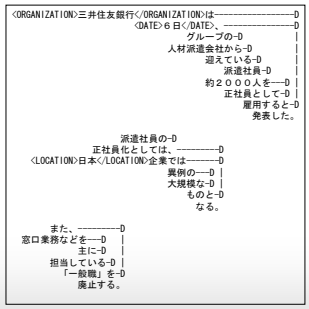
三井住友銀行は6日、グループの人材派遣会社から迎えている派遣社員約2000人を正社員として雇用すると発表した。派遣社員の正社員化としては、日本企業では異例の大規模なものとなる。また、窓口業務などを主に担当している「一般職」を廃止する。すべての社員に幹部社員への道を開くことで意欲を高めるとともに優秀な人材を確保するのが狙いで、一般職の廃止はメガバンクでは初めて。

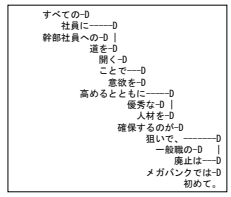
いずれも来年7月1日から実施する。正社員化される派遣社員は来夏以降新設する「ビジネスキャリア職」になり、主に事務職を担当する。現在の一般職は、ビジネスキャリア職か地域限定の個人向け営業職員である「コンシューマーサービス（CS）職」のいずれかを選択できる。（以下略）

## 長文の構文解析 syntactic parsing for long sentences

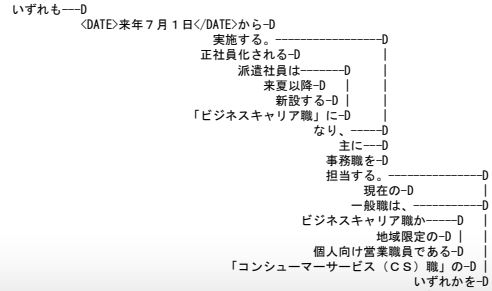



## 長文の構文解析 syntactic parsing for long sentences




## 長文の構文解析 syntactic parsing for long sentences




## 論文紹介と実験の紹介

**WWWテキストの自動要約とKWICインデックスの作成**
- 清田陽司、黒橋禎夫、情報処理学会、自然言語処理研究会、2000
  - 自動要約によってWWW テキストへのKWIC (Key Word In Context) インデックスを作成
  - KNPを利用して，文を単文や句の形で要約

⇒ これを試した

## システム構成図

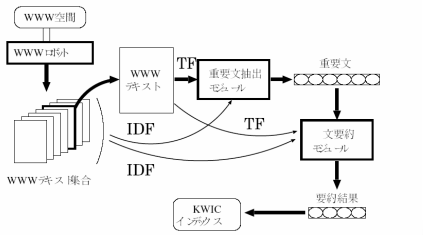



図1: システム構成図

## 方法① キーワードのスコアを求める

- 記事の各文をJUMANで形態素解析を行い、品詞分類が以下であるものをキーワード候補として抽出
  - 普通名詞、サ変名詞、固有名詞、地名、人名、組織名、カタカナ、アルファベット（今回は簡単のためこれだけにした）
- TF-IDF手法を用いてキーワードのスコアを計算

## 結果（スコアの上位10単語）

- エンターテインメントの記事

en1：エンターテインメントの記事①

| | TFij | Nj | w(i,j) |
|---|---|---|---|
| 菌 | 30 | 1 | 41.4 |
| しもん | 6 | 1 | 8.3 |
| もや | 6 | 1 | 8.3 |
| カビ | 5 | 1 | 6.9 |
| 石川 | 5 | 1 | 6.9 |
| アニメ | 4 | 1 | 5.5 |
| マンガ | 4 | 1 | 5.5 |
| 白 | 4 | 1 | 5.5 |
| 組 | 4 | 2 | 4.3 |
| 登場 | 4 | 3 | 3.6 |

en2：エンターテインメントの記事②

| | TFij | Nj | w(i,j) |
|---|---|---|---|
| デップ | 8 | 1 | 11.0 |
| ジョニー | 5 | 1 | 6.9 |
| 犯罪 | 4 | 1 | 5.5 |
| マン | 5 | 3 | 4.5 |
| 決定 | 3 | 1 | 4.1 |
| 監督 | 4 | 3 | 3.6 |
| アメリカ | 3 | 2 | 3.2 |
| ディリンジャー | 2 | 1 | 2.8 |
| プロジェクト | 2 | 1 | 2.8 |
| マイケル | 2 | 1 | 2.8 |

en3：エンターテインメントの記事③

| | TFij | Nj | w(i,j) |
|---|---|---|---|
| 志郎 | 7 | 1 | 9.7 |
| 清 | 7 | 1 | 9.7 |
| 復活 | 7 | 1 | 9.7 |
| ジョン | 7 | 2 | 7.6 |
| ヨーコ | 5 | 1 | 6.9 |
| がん | 4 | 1 | 5.5 |
| 活動 | 4 | 2 | 4.3 |
| ベイベー | 3 | 1 | 4.1 |
| レノン | 3 | 1 | 4.1 |
| 館 | 3 | 1 | 4.1 |

TFij ： キーワード kj のテキスト Di での出現回数　　N ： テキスト数（=24）

Nj ：キーワード kj の出現するドキュメント数

w(i,j)：テキスト Di に対するキーワード kj のスコア　　$w(i,j) = TFij \times \log ( N / Nj )$

## 方法② 重要文の選択

- 各テキストについて、以下の式より重要文を選択

文のスコア　$I(i,t) = \frac{\sum_{j=1}^{m_t} w(i,j)}{(m_t)^n}$

mt：文に含まれるキーワード数
w(i,j)：キーワードのスコア
n=0.5とした

  - wwwテキストにおいては経験的に重要文が存在する範囲は以下に限定
    - 箇条書き属性をもつ文とc(=15)文字以下の文を除く先頭よりa(=6)文

## 重要文の選択（結果の例）

- キーワードの重要度スコアを用いて、各文の重要度スコアを計算

文1：スコア　27．3

文のスコア：　$I(i,t) = \frac{\sum_{j=1}^{m_t} w(i,j)}{(m_t)^n}$

＜特集＞大ヒット「もやしもん」 カワイイ「菌」がをかもす

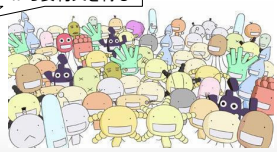
文2：スコア　31．6

目には見えない極小の「菌」が見え、言葉が交わせる農大生とさまざま個性を持った菌やウイルスが繰り広げるドタバタ劇を描いた石川雅之さんのマンガ「もやしもん」。

文3：スコア　21．9

「かもすぞ！」と叫ぶカワイイ菌たちが評判となり、コミックスも売り切れが続出、10月からフジテレビのアニメ枠「ノイタミナ」で放送が始まったアニメも好調だ。

文2を重要文とする
ここから要約文を得る



「もやしもん」に登場する100種類以上のかわいい「菌」たち

## 方法③ 要約文の生成

- 構文木の分割
  - 構文解析された文を小さなパート（単文、名詞句）に分割する
    各パートは、構文木上で隣接している限りどのように結合しても文として意味をなすと考えられる

分割の規則
- 連用節の分割
- 連体節の分割
- デ格の分割
- 副詞、接続詞の分割
- 時間を表す名詞句の分割
- 並列要素の分割
  - 末尾省略可能表現

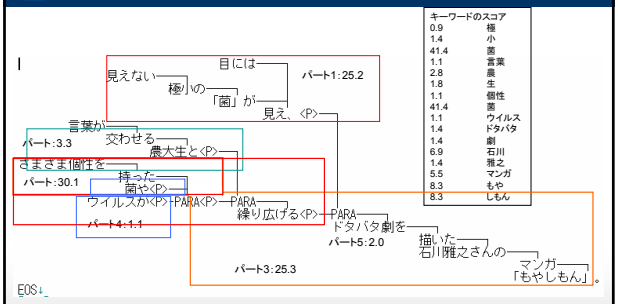
左に挙げたルールに加え、
例外がいくつかある

## 方法③ 要約文の生成

- 各パートの重要度スコアを計算

パートのスコア $S(i) = \dfrac{\sum_{j=1}^{k(i)} w(i,j)}{k(i)^n}$

## 構文木の分割例




さまざま個性を持った菌が繰り広げるドタバタ劇を描いた石川雅之さんのマンガ「もやしもん」

## 補足①

- その他に論文で書いてあったこと(こうしたほうが精度が上がる)
  - HTML属性にもとづくキーワードの重み付け
    - Title、見出し(H1~H6)
  - 文末のパートの重要度スコアをm(=2)倍

表:重要文抽出モジュールの評価結果

| | | |
|---|---|---|
| 正解 | 149 (74.5%) | 82 (41.0%) |
| 不正解 | 51 | 118 |

表:文要約の評価結果

| | 制限文字数 | |
|---|---|---|
| | 25文字 | 45文字 |
| ○ | 109 | 153 |
| △ | 68 | 34 |
| × | 23 | 13 |

## 補足②

- 誤りの要因
  - 構文解析の間違い(50%)
  - TF-IDFによる重み付けがうまくいかない(30%)
  - 文字数制限のため重要な情報がいれられない(20%)

表:重要文抽出モジュールの評価結果

| | | |
|---|---|---|
| 正解 | 149 (74.5%) | 82 (41.0%) |
| 不正解 | 51 | 118 |

表:文要約の評価結果

| | 制限文字数 | |
|---|---|---|
| | 25文字 | 45文字 |
| ○ | 109 | 153 |
| △ | 68 | 34 |
| × | 23 | 13 |