社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－NL－180 (22)
2007／7／25

# サンプリング技術を利用した文章類似性評価

山田一郎† 中田洋平‡ 松井淳†‡ 松本隆‡
三浦菊佳† 住吉英樹† 八木伸行†

†NHK放送技術研究所 〒157-8510 東京都世田谷区砧 1-10-11
‡早稲田大学大学院理工学研究科〒169-8555 東京都新宿区大久保 3-4-1
E-mail: yamada.i-hy@nhk.or.jp

**あらまし** テレビ番組のナレーションでは、「場所紹介」や「人物紹介」など特定の事柄を表現するために同じような言い回しが多用される。このような言い回しを含む文章区間が抽出できれば、対応する番組映像区間の場所紹介や人物紹介といったメタデータを付与することができる。本稿では、番組のクローズドキャプションから特定の事柄を表現する文章に類似した文章を抽出するために、文章間の類似性を評価する手法を提案する。提案手法では文章を構文解析した結果、得られる木構造中の部分木を特徴とし、この特徴をサンプリングして学習する GibbsBoost アルゴリズムを用いて文章間の類似性を評価する。紀行番組のクローズドキャプションを対象として、場所を映像とともに説明する定型表現文章区間にある文章との類似性を評価する実験を行い、提案手法の有効性を確認した。

**キーワード** メタデータ生成，特定表現抽出，木構造解析，ギブスブースト，サンプリング

# Evaluation of the Similarity between Multiple Sentences using Sampling Techniques

Ichiro YAMADA† Yohei NAKADA‡ Atsushi MATSUI†‡ Takashi MATSUMOTO‡
Kikuka MIURA† Hideki SUMIYOSHI† and Nobuyuki YAGI†

† NHK Science & Technical Research Laboratories 1-10-11 Kinuta, Setagaya-ku, Tokyo, 157-8510 Japan
† Dept. of Electrical Engineering and Bioscience, Waseda University 3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555 Japan
E-mail: yamada.i-hy@nhk.or.jp

**Abstract** In the closed captions, there are a lot of typical expressions to express specific things, for example, first introduction of a guest in a talk show or explanation of a place in travel program. Such information helps us to put metadata to the corresponding scenes. This paper proposes a method to evaluate the similarity between multiple sentences in order to extract a section in which sentences are similar to the typical expressions expressing specific things. The first step generates tree structures from input section of sentences and extracts subtrees from these tree structures. We use Gibbsboost algorithm which samples these subtrees for features and learns the features to evaluate the similarity. In the experiment of judging whether a section of sentences is similar to the section which explains a place with video targeting closed captions of TV programs concerned with travel, we show the effectiveness of our method.

**Keyword** Metadata generation, Typical expression extraction, Tree Structure analysis, GibbsBoost Algorithm, sampling

## 1. はじめに

近年、放送局では番組を蓄積・管理するシステムが普及し、NHK においても NHK アーカイブスに約 61 万番組が蓄積されている[1]。これらのコンテンツを有効活用するためには、番組のどの区間に何が映っているかというメタデータが重要な役割を果たす。我々は、

番組のクローズドキャプションを処理対象として番組映像にメタデータを自動付与する研究に取り組んでいる[2][3]。テレビ番組では、「場所紹介」や「人物紹介」など特定の事柄を表現するために同じような言い回しが多用される。例えば、表1に示すクローズドキャプションでは、矩形で囲まれた部分が「場所」を映像とともに説明している。最初に体言止めにより「オンフルール」という町の位置情報を説明し、次に町の詳細を断定の助動詞「です」を使って説明している定型的な表現である。このような文章区間を抽出することにより、対応する番組映像区間に映像内容のメタデータを付与することができる。

特定事項を表現する文章区間抽出のために複数文を処理対象とする場合、得られる特徴は膨大な量となる。本稿では、特徴をサンプリングして学習するGibbsBoostアルゴリズム[4]を用いて、特定の事柄を表現する言い回しを含む文章との類似性評価手法を提案する。文章から抽出する特徴に対して、先見的情報によりサンプリングを行う。学習処理においても、複数の弱学習器の系列を生成し、逐次モンテカルロ法[5]によりサンプリングして解を導くことで、効率的かつ効果的な処理を実現する。中田らにより発案され研究が進められているGibbsBoostアルゴリズム[4]は、これまで画像処理による顔画像検出に適用された[6]。今回、このアルゴリズムを自然言語処理に適用することにより、自然言語処理におけるサンプリング技術の有効性を検証する。

以下、2章では類似性評価に関する関連研究について述べる。3章ではGibbsBoostアルゴリズムの説明を行い、4章ではGibbsBoostアルゴリズムを類似文章検索に応用する手法について説明する。5章では実際のクローズドキャプションを利用した類似性評価実験を行い、最後にまとめと今後の課題について言及する。

## 2. 関連研究

文と文の類似性を評価する手法として Collins らにより Tree Kernel が提案されている[7]。この手法では、テキストに含まれる共通部分木の数により類似性を評価しているが、部分木は膨大な数となるため処理速度の問題があげられている。そこで、市川らは Tree Kernel を近似する高速処理可能な手法を提案した[8]。また、工藤らは部分木を素性とする decision stumps[9]とそれを弱学習器とした boosting アルゴリズムを提案し、製品レビュー文や新聞記事のテキスト分類の実験を行っている[10]。これらの部分木を特徴として利用する手法では、ノードの飛び越えを許さない部分木の完全一致を類似度判定の基準としているため、結果として局所的な部分木しか特徴として利用されないことが多い。また、複数文にまたがる類似性評価は行われていない。

我々はこれまでに、ノードの飛び越えを許した部分木を利用して木構造間の類似度を利用した弱学習器を生成し、AdaBoost アルゴリズム[11]による学習を行うことで、文章間の類似性を評価する手法を提案している[2]。ノードの飛び越えを許すことにより、構文木で遠く離れて位置する文節間の特徴なども考慮した類似性が評価でき、さらには、複数文を対象とした文集合の類似性評価も可能となる。類似性評価の処理では、木構造から生成した大量の部分木を基として弱学習器を生成する。この際、比較対象となる木構造から生成される部分木の数は膨大になるうえ、最終的には類似性評価に使われない無駄な部分木も弱学習器として生成するため効率が悪いという問題が残されていた。

提案する手法では、抽出された部分木に対して重要性の事前分布を定義し、弱学習器を生成する際には事前分布を基とするサンプリングを行う。学習においては、弱学習器の系列を複数生成し、実際のデータに対する適応度を示す提案分布を基とするサンプリングを行う。2種類のサンプリング処理を行うことにより効率的な処理が可能となるとともに、弱学習器の系列を複数生成した処理により頑健性が向上する。

## 3. GibbsBoost アルゴリズム

Boosting アルゴリズムは教師有り学習手法の一つで、2値の出力を持つ学習データが与えられたとき、逐次的に学習データの重みを変化させながら誤り率を最小化する弱学習器が選択される。この弱学習器を組み合

表1 クローズドキャプション例（矩形で囲まれた部分は「場所」を説明する定型的な表現区間）

| 提示時間 | クローズドキャプション |
|---|---|
| 08:29:03 | 絵は 全然描きませんからって→ |
| 08:29:09 | まっ こんなとこですかね。 |
| 08:29:12 | やっぱり 絵を描かなくてよかったかもしれませんね。 |
| 08:29:46 | セーヌ川を挟み ル・アーブルの対岸に位置する港町 オンフルール。 |
| 08:29:53 | 今なお中世の古い家並みが残る 町です。 |
| 08:29:59 | 18歳の時 モネは パリに出て画家を 目指しますが 美術学校の 入学試験に合格しませんでした。 |
| 08:30:11 | 実家に戻る事を 強要した父親の意向に反して なおも パリにとどまって絵の勉強を し続けた モネ。 |

わせて精度の高い学習器を構成し、観測データが2値のいずれであるかを判別する。判別関数は式(1)で表される。

$$F(x;\Theta_t) = \sum_{t'=1}^{t} \alpha_{t'} h(x;\theta_{t'}) \qquad (1)$$

ここで、$x$ は観測データ、$h$ は弱学習器、$\alpha_t$ は $t$ 番目の弱学習器の信頼度、$\theta_t$ は $t$ 番目にどの弱学習器を選択するかを決めるパラメータである。また、$\Theta_t := (\alpha_1,\ldots,\alpha_t,\theta_1,\ldots,\theta_t)$ とする。学習処理では、入力 $x_i$ と出力 $y_i$ からなる学習データ $\{y_i,x_i\}_{i=1}^N$ が与えられた時、損失関数 $\sum_{i=1}^N L(y_i F(x;\Theta_t))$ を最小化する $\Theta_t$ を $t=1,\ldots,T$ について逐次的に求める。$\Theta_t$ の決定により、入力 $x$ に対する判別関数 $F$ の値が0より大きい場合+1を返し、0以下の場合-1を返す2値判別器を実現できる。

GibbsBoostアルゴリズムもBoostingアルゴリズムの一つであり、弱学習器を効果的かつ効率的にサンプリングすることで弱学習器の系列を複数生成する。GibbsBoostアルゴリズムでは、Boostingアルゴリズムの損失関数に対応するエネルギー関数 $L(z)$ を用いて、パラメータ $\Theta_t$ に対する確率分布 $P_t(\Theta_t)$ を(2)式により定義する。

$$P_t(\Theta_t) \propto \pi(\Theta_t)\prod_{i=1}^{N} \exp\left(-\beta_t L(y_i \frac{F(x_i;\Theta_t)}{\sqrt{t}})\right) \qquad (2)$$

式(2)では、損失関数 $L(z)$ からの影響が $t$ に比例して増えないよう $\sqrt{t}$ により抑制している。$\beta_t$ は、統計力学における温度の逆数を示す係数であり、確率分布 $P_t(\Theta_t)$ の分散を抑制する。$\beta_t$ が大きい場合、$\Theta_t$ は損失関数の和が小さくなるような値に集中し、$\beta_t$ が小さい場合、$\pi(\Theta_t)$ と似た分布となる。本実験では式(3)に示すBoltzmann annealing[12]をベースとした値と、式(4)に示すCauchy annealing[12]をベースとした値を利用する。

$$\beta_t = \beta_0 \log(t+e) \qquad (3)$$

$$\beta_t = \beta_0 (t+1) \qquad (4)$$

$\pi(\Theta_t)$ はパラメータ $\Theta_t$ に対する事前確率分布であり、式(5)により定義される。

$$\pi(\Theta_t) = \prod_{t'=1}^{t} \pi_\theta(\theta_{t'})\pi_\alpha(\alpha_{t'}) \qquad (5)$$

$\pi_\theta(\theta_{t'})$ は、$t'$ 番目の弱学習器の候補を選択する事前分布であり、先見的情報に基づいて決定する。$\pi_\alpha(\alpha_{t'})$ は $t'$ 番目の弱学習器の信頼度に対する事前分布であり、ここでは正規分布と定める。$\pi(\Theta_t)$ の詳細は4章で説明する。

GibbsBoostアルゴリズムにおける判別関数は、使用する弱学習器の系列数を $T$ 個としたとき、$F(x;\Theta_T)$ に対してパラメータ $\Theta_T$ に対する確率分布 $P_T(\Theta_T)$ による期待値により、式(6)で定義される。

$$F_{ave,T}(x) = \int F(x;\Theta_T) P_T(\Theta_T) d\Theta_T \qquad (6)$$

式(6)は、解析的に解を求めることが困難である。そこで逐次モンテカルロ法を利用し、$\Theta_T$ を有限個数だけサンプリングする。サンプリングの処理手順を図1に示す。逐次モンテカルロ法では、パラメータ $\Theta_t$ のサンプリングと、その重み(importance weight)の計算のために提案分布 $Q$ を使用する。$Q$ の値が大きい部分から多くサンプリングし、$Q$ の値が小さい部分からはあまりサンプリングを行わない。そして、$Q$ をもとに取り出した $M$ 個のサンプル $\Theta_t^{(j)}$, $j=1\sim M$ に対する重み $w^{(j)}$ (importance weight)を計算する。この重み $w^{(j)}$ は、選ばれた弱学習器の系列に対する $P_t(\Theta_t)$ の値により算出される。この重み $w^{(j)}$ の値を利用して、選ばれた $\Theta_t^{(j)}$ からリサンプリングを行う。大きな重みを持つ $\Theta_t^{(j)}$ は何度も選択され、小さな重みを持つ $\Theta_t^{(j)}$ は選択されない。この処理を、線形結合する弱学習器数 $t=1\sim T$ のそれぞれの場合において逐次的に行うことにより、$T$ 個からなる弱学習器系列が $M$ 個生成される。これらの系列がサンプリングされた結果となる。

提案分布 $Q(\alpha_t,\theta_t;\Theta_{t-1})$ は、弱学習器の選択を決定する分布 $Q(\theta_t)$ と、選ばれた弱学習器の信頼度を決定する分布 $Q(\alpha_t;\theta_t,\Theta_{t-1})$ の積 $Q(\alpha_t,\theta_t;\Theta_{t-1}) = Q(\alpha_t;\theta_t,\Theta_{t-1})Q(\theta_t)$ で表現できる。$Q(\theta_t) = \pi_\theta(\theta_t)$ とし、$Q(\alpha_t;\theta_t,\Theta_{t-1})$ は

弱学習器の各系列数( $t$ = 1 to $T$ )において、Step1～Step3を繰り返す

Step1: $t$-1 個の弱学習器が線形結合された系列 $j$ ( $j$ = 1～$M$)に対して、$t$ 番目の弱学習器を提案分布 $Q$ に基づきサンプリング

$$(\alpha_t^{(j)},\theta_t^{(j)}) \sim Q(\alpha_t,\theta_t;\Theta_{t-1}^{(j)})$$

Step2: Step1 において選択された系列 $\Theta_t^{(j)}$ の Importance weight $w^{(j)}$ を計算

$$w^{(j)} \propto \frac{P_t(\Theta_t^{(j)};\beta_t)}{P_{t-1}(\Theta_{t-1}^{(j)};\beta_{t-1})Q(\alpha_t^{(j)},\theta_t^{(j)};\Theta_{t-1}^{(j)})}$$

ここで、$\sum_{j=1}^M w^{(j)} = 1$

Step3: 確率 $w^{(j)}$ によって $\Theta_t^{(j)}$ をリサンプリング

図1. GibbsBoostアルゴリズムにおけるサンプリング処理手順

学習データに対するエラーレートから算出する。

$M$ 個のサンプリング結果を利用して、$F_{avs,T}(x;\Theta_T)$ の和を計算する。$M$ 個のサンプルを $\{\Theta_T^{(j)}\}_{j=1}^{M}$ とすると、式(6)は式(7)で近似される。

$$F_{ave,T}(x) \approx \frac{1}{M}\sum_{j=1}^{M} F(x;\Theta_T^{(j)}) \qquad (7)$$

この値の正負を判別基準とすることにより、2 値判別が可能となる。

## 4. 文章類似性評価処理

本処理では与えられた複数文からなる文章が、特定の事柄を表現するための定型表現文章区間と類似しているか否かを判別する。まず、複数文からなる文章から構文解析結果の木構造を抽出し、次に、木構造間の類似性を評価指標とする弱学習器を生成する。GibbsBoost アルゴリズムにより、生成された弱学習器をサンプリングし、式(7)の関数を生成する。以下に部分木抽出、弱学習器生成、そして、GibbsBoost による学習について記す。

### 4. 1. 部分木抽出

入力テキストを一文ごとに構文解析して、各ノードを文節により構成する構文木を生成する。各文の根ノードの親ノードに最上位ノードを生成し、最上位ノードから各文の構文木へは順序付きのアークで結んだ木構造を生成する。順序付きアークは文の出現順序を考慮した木構造間の類似度評価で利用する。表 1 の矩形で囲まれた区間の入力テキストを木構造に変換した例を図 2 に示す。次に、学習データ中の正例として与えられた木構造からキーとなる単語を含む部分木を生成する。今回の実験では「場所」を映像とともに説明する言い回しの有無を対象としているため、「オンフルール」のような地名を表す単語は「地名」、「町」のような地名の言い換え表現を表す単語は「地言換」というラベルで抽象化した。また、部分木の作成の際にノードの飛び越えを許し、飛び越えたノードは"+"の記号で置き換え 1 つ以上のノードとのマッチングを許した。図 2 に示した木構造から生成される部分木の一部を図 3 に示す。

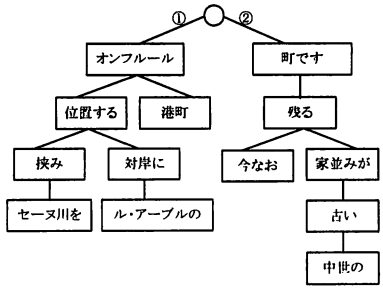



図 2. 木構造生成例

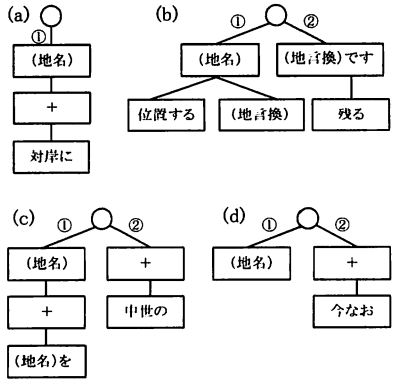



図 3. 木構造から抽出された部分木（一部）

### 4. 2. 弱学習器生成

抽出した部分木と、学習データに含まれるテキストから生成される木構造との類似度は、部分木に含まれる葉ノードから根ノードまでの全リスト構造を抽出し、その各リスト構造が対象とする木構造に含まれる割合を基準として定義する。部分木 $r$ と木構造 $x$ の類似度 $sim(r,x)$ は式(8)とする。

$$sim(r,x) = \frac{1}{N(r)}\sum_{r_i \in r}\frac{1}{K(r_i)}\sum_{sr \in r_i}\max_{sx \in x}(C^d \times sim'(sr,sx)) \qquad (8)$$

$r_i$：部分構造 $r$ に含まれる $i$ 番目の文
$sr$：$r_i$ に含まれる葉ノードから根ノードまでのリスト
$sx$：$x$ に含まれる葉ノードから根ノードまでのリスト
$sim'(sr,sx)$：$sr$ が $sx$ に含まれる割合。リストに含まれる主辞と付属語を分割して計算。
$N(r)$：$r$ に含まれる文数
$K(r_i)$：$r_i$ に含まれるリスト数
$C$：文の出現順序の差に与えるペナルティ値（本実験では 0.5 とした）
$d$：文の出現順序の差

文の出現順序の差は、複数文からなる構文木と部分木を比較するときに生じるもので、複数の組み合わせの可能性がある場合は、その最大値を $sim(r,x)$ とする。類似度が一定値以上か否かを判断基準とすることにより、入力 $x$ に対して部分木 $r$ と閾値 $\varphi_r$ を変数に持つ弱学習器 $h(x;r,\varphi_r)$ を生成する。

### 4.3. GibbsBoost による学習

部分木生成時にキーとなる地名以外にいくつのノード（文節）を利用するかにより、弱学習器の特徴が決まる。しかし、利用するノード数が多い場合は計算量も膨大となる。もし、100 個のノードから 50 個抽出する場合、その組み合わせ数は $10^{29}$ 個を超え計算が困難と考えられる。そこで、効率的な部分木のサンプリングが必要となる。

図2などに示されるような構文木では、各文の根ノードに主節の述部があり、その下のノードには、主節の述部に直接係る連体節または主節の格要素が位置する。これらは文を比較する上で重要な要素と考えられる。そこで、根ノードに近いノードほど選択される確率が高くなり、さらに、選択されたノード間の距離が近いほど選択される確率が高くなるような事前分布 $\pi_\theta(\theta)$ を式(9)の通り定義する。

$$\pi_\theta(\theta) \propto \prod_{n\in\theta} C_1^{depth(n)} \times \prod_{n_1,n_2\in\theta,n_1\neq n_2} C_2^{length(n_1,n_2)} \quad (9)$$

$depth(n)$：ノード $n$ の根ノードからの深さ
$length(n_1,n_2)$：ノード $n_1$ とノード $n_2$ 間のノード数
$C_1$：ノードの深さに対するペナルティ(本実験では0.9)
$C_2$：2つのノード間の距離に対するペナルティ(本実験では0.95)

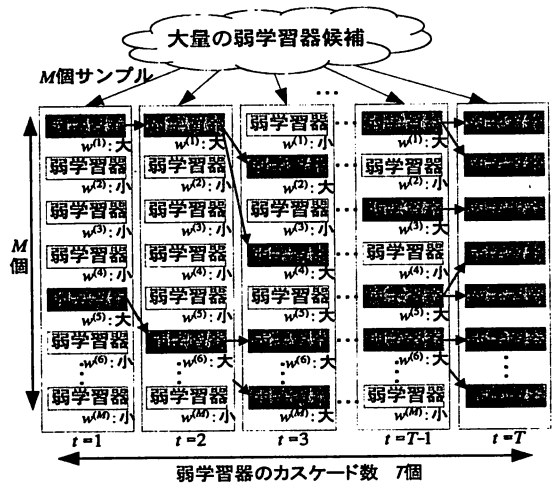
学習データの正例の構文木から生成された大量の部分木（弱学習器）に対して、式(9)の値により一定数 $M$ 個をサンプリングし(図1 Step1)、次に選択された $M$ 個に対して Importance weight を計算する(図1 Step2)。この Importance weight の値によって、利用する弱学習器を決定する。Importance weight の値が高い系列ほど、次の処理でも利用される確率が高くなる。さらに、改めて $M$ 個の弱学習器をサンプリングし、同様の処理を $T$ 回繰り返すことにより、$T$ 個の弱学習器がカスケードした系列が $M$ 個出来る。図4に GibbsBoost アルゴリズムの概念図を示す。ここでは、網掛けの弱学習器が最終的に選択された弱学習器となっている。この系列を利用して、最終的に、式(7)により2値判別を行う。

## 5. 「場所」を説明する定型表現文章区間の判別実験

NHK で放送された紀行番組「わが心の旅」のクローズドキャプションを対象として、入力文章が、「場所」を映像とともに説明している定型的な文章と、場所を表す単語があるが場所を映像とともに説明していない文章のどちらに類似しているか判別する実験を行った。48番組に対して人手により「場所」を映像とともに説明している定型的な文章154区間を抜き出し学習データの正例とした。負例も、正例と同数だけ人手により無作為に抽出した。この学習データを2つに分け、一方を学習データ、他方をテストデータとしたクロスバリデーションによる実験を行った。使用するノード数を3個とした時、1回の試行では、平均10655個の弱学習器が生成された。以下にクローズドデータによる最適パラメータ値の推定処理と、推定されたパラメータを利用した実験について記す。

### 5.1. 最適パラメータ値の推定処理

弱学習器の変数 $\Theta_t$ に対する確率分布 $P_t(\Theta_t)$ として利用した式(2)の Gibbs 方程式において係数 $\beta_t$ が存在する。この係数 $\beta_t$ により、どの弱学習器が選択されるか変動する。また、サンプリングする弱学習器の数と弱学習器の系列の長さも精度に影響する。実験で使用するこれらのパラメータの最適値を推定するため、学習データをテストデータとしたクローズドテストを行




$w^{(i)}$ :$i$ 番目の弱学習器の Importance weight
$M$:1回の弱学習器サンプリング数

図4. GibbsBoost アルゴリズムの概念図

表2 Boltzmann annealing による判別結果の正解率(平均)

| | M=500 T=500 | M=1000 T=1000 |
|---|---|---|
| $\beta_0$=5.0 | 139.5/154 (90.6%) | 142.1/154 (92.3%) |
| $\beta_0$=10.0 | 134.8/154 (87.5%) | 144.0/154 (93.5%) |
| $\beta_0$=15.0 | 141.5/154 (91.9%) | 142.6/154 (92.7%) |
| $\beta_0$=20.0 | 140.9/154 (91.5%) | 140.8/154 (91.4%) |
| $\beta_0$=30.0 | 143.5/154 (93.2%) | 147.9/154 (96.0%) |

表3 Cauchy annealing による判別結果の正解率(平均)

| | M=500 T=500 | M=1000 T=1000 |
|---|---|---|
| $\beta_0$=0.2 | 143.4/154 (93.1%) | 145.5/154 (94.5%) |
| $\beta_0$=0.5 | 137.8/154 (89.5%) | 145.4/154 (94.4%) |
| $\beta_0$=0.7 | 144.8/154 (94.0%) | 148.2/154 (96.2%) |
| $\beta_0$=1.0 | 141.5/154 (91.9%) | 145.2/154 (94.3%) |
| $\beta_0$=1.5 | 141.6/154 (91.9%) | 147.4/154 (95.7%) |

った。実験ではサンプリング時の乱数の影響を吸収するために 5 回の試行を行った。Boltzmann annealing ($\beta_0$={5.0, 10.0, 15.0, 20.0, 30.0})と Cauchy annealing ($\beta_0$={0.2, 0.5, 0.7, 1.0, 1.5} )、サンプリングする弱学習器の数 $M$={500, 1000}、弱学習器の系列の長さ $T$={500,1000}としたときの判別結果の正解率を表 2 と表 3 に示す。

実験の結果、サンプリングする弱学習器の数 $M$=1000、弱学習器の系列の長さ $T$=1000 で $\beta_0$=0.7 の場合の Cauchy annealing の正解率が最良であった。

## 5. 2. 推定されたパラメータを利用した実験

前節で求められたパラメータの最適値を使用して、学習データとは異なるテストデータを利用した文章判別実験を行った。パラメータ値推定処理と同様に、サンプリング時の乱数の影響を吸収するために 5 回の試行を行った。

比較対象として、我々がこれまでに提案している AdaBoost を利用した手法[2]と、ベクトル空間モデルを利用した手法[13]による実験を行った。AdaBoost を利用した手法では、弱学習器生成までは同じ処理を行い、重み付きの学習データに対する誤り率を最小とする弱学習器を全ての弱学習器の中から選択する。弱学習器の系列の長さを $T$=1000 とした実験を行った。ベクトル空間モデルを利用した手法では、文章に含まれる単語に対して TFIDF 値により重み付けをしたベクトルを作成し、テストデータから生成したベクトルが、学習データにある正例文章から生成したベクトルと負例文章から生成したベクトルのどちらに類似しているかを、そのコサイン距離により判定する。場所説明をしている定型表現文章区間か否かを判別した実験の正解率を表 4 に示す。

表 4　場所説明をしている定型区間判別結果

| | 定型的な文章 | 定型的でない文章 | 全体 |
|---|---|---|---|
| GibbsBoost (提案手法) | 138.4/154 (89.9%) | 140.2/154 (91.0%) | 278.6/308 (90.5%) |
| AdaBoost | 125/154 (81.2%) | 148/154 (96.1%) | 273/308 (88.6%) |
| ベクトル空間モデル | 148/154 (96.1%) | 103/154 (66.9%) | 261/308 (84.7%) |

Xeon™ 2.80GHz x 2 の PC を用いた各手法における平均学習時間(CPU Time)は、以下の通りであった。

| | |
|---|---|
| GibbsBoost (提案手法) | 14 分 11 秒 |
| AdaBoost | 650 分 34 秒 |

GibbsBoost を利用した提案手法は、AdaBoost を利用した手法やベクトル空間モデルを利用した手法と比べて全体の正解率で上回り、さらに学習時間は AdaBoost を利用した従来手法と比べて 45 倍以上の速さを実現していることがわかる。

## 6. まとめ

本稿では、GibbsBoost アルゴリズムを用いて、特定の事柄を表現するための言い回しを含む文章か否かを判別する手法を提案した。実験では、従来の AdaBoost を用いる手法やベクトル空間モデルを用いる手法と比べて高精度に分別可能であることを示し、学習時間を大幅に短縮できることが確認できた。統計手法を利用した自然言語処理では、膨大な量の言葉の特徴を必要とするケースが多く、特徴の事前分布を利用したサンプリング処理は、有効な手段となりうる。今後は、処理対象を広げた文章検索の実験を進めるとともに、サンプリング技術を他の自然言語解析処理に応用していく予定である。

## 文　献


[1] NHK アーカイブス (http://www.nhk.or.jp/nhk-archives/)

[2] 山田, 三浦, 住吉, 八木, 奥村, 徳永: AdaBoost を利用した字幕テキストからの定型表現文章区間抽出, 情処学会研究報告 NL174, Vol.2006, No.82, pp25-30(2006)

[3] 三浦, 山田, 住吉, 八木: クローズドキャプションを利用した映像主被写体の推定手法, 情処学会研究報告 NL171-1, Vol.2006, No.1, pp1-6(2006)

[4] Y. Nakada, Y. Mouri, Y. Hongo, T. Matsumoto: Gibbsboost: a Boosting Algorithm using a Sequential Monte Carlo Approach, Proceedings of the 2006 16th IEEE Signal Processing Society Workshop, pp259-264(2006)

[5] A. Doucet et. al.: Sequential Monte Carlo Methods in Practice. Springer(2001)

[6] 木村, 松井, 中田, 松本: GibbsBoost による正面顔画像検出:事前情報を考慮する Bayes 的アプローチ, 第 5 回情報科学技術フォーラム FIT2006, I-008, pp.17-18(2006)

[7] M. Collins, and N. Duffy, Convolution Kernels for Natural Language, In Proceedings of NIPS2001 (2001)

[8] 市川, 橋本, 徳永, 田中: テキスト構文構造類似度を用いた類似文検索手法, 情処学会研究報告 FI-079, Vol.2005, No.42, pp39-46(2005)

[9] R.E. Schapire, and Y. Singer: BoosTexter: A boosting -based system for text categorization. Machine Learning, 39(2/3), pp135-168(2000)

[10] 工藤, 松本: 半構造化テキストの分類のためのブースティングアルゴリズム, 情処論文誌, Vol.45, No.9, pp2146-2156(2004)

[11] Y. Freund and R.E. Schapire: A decision theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences, Vol.55, No.1, pp.119-139(1996)

[12] H. Szu and R. Hartley: Fast simulated annealing, Physics Letters A, 122, pp.157-162(1987)

[13] G. Salton and C. Buckley: Term-weighting approachs in automatic text retrieval, Proc. Information Processing & Management, 24(5), pp513-523(1988)