被験者判定のゆれと要約モデル

野本忠司[†] 松本裕治[††]

本稿では，人間の重要文判定データを教師あり（supervised）・教師なし（unsupervised）の学習パラダイムのもとでモデリングし，データの性質と特定のモデルの精度との関係について考察する．具体的には，確率的決定木をベースにした判定データを直接学習する手法と，判定データをまったく参照しないクラスタリングに基づく手法とを比較し，両者の精度と判定者間の一致度との関係を見る．実験の結果，以下の点が確認された．(a) クラスタリング手法がおおむね決定木手法に比べてパフォーマンスが良い．(b) クラスタリング手法は判定者間のゆれの影響をあまり受けないが，決定木手法は性能がゆれに左右される．(c) いずれの手法も性能が文章構造に強く影響される．

Exploiting Human Judgments for Automatic Text Summarization: An Empirical Comparison

Tadashi Nomoto[†] and Yuji Matsumoto[††]

The paper empirically examines how variation in human judgments on sentence extraction affects performance of summarizers. In particular we will be concerned with how summarizers from two different learning paradigms, i.e., supervised v. unsupervised paradigms, fare when set to the task of extracting a summary from a text. We build a supervised summarizer on the probabilistic decision tree and an unsupervised summarizer on the K-means clustering, and compare performance of the two approaches, and some variation on them, on data elicited from human subjects. It is found that for the most of the time, the clustering approach outperforms the supervised approach. Somewhat to our surprise, we also found that the variability in judgment exerts no significant effect on how well the clustering based approach performs, in contrast to the supervised approach, which is hurt by the variability. Another notable result is that what we might call the topical structure of text apparently influences performance of the summarizers, whether supervised or unsupervised.

1. はじめに

一般に重要文抽出タスク☆における人間の重要文判断は，個人間のゆれが著しく，統一的な選択基準を定義することが困難であることがよく知られている[8),14),18),24)]．過去の報告例からみても，重要文判定における判定者間のゆれはきわめて一般的であることから，タスクのデザインというより，むしろ人間の判定行為の 1 つの特性と考えるのが自然であり，むしろ積極的に研究の対象にすべきであると思われる．しかし，従来の要約研究の多くはゆれを単なる雑音として，あるいは，あってはならないものとして排除する傾向が強く，自然現象としてみたときのゆれと要約モデルとの関係についての研究はまだ例がない[2),4),11),15),28)]．

このような背景の下，本稿はデータ・サイエンス的な観点から，いくつかの異なったアプローチによる要約モデルを導入し，どのようなアプローチがゆれを含む（あるいは含まない）人間の判定データをよく予測できるか明らかにしようというものである．特に，本稿では文章がユニークな要約を持たないとき，ゆれがあるということにする．また,「人間の判定データ」とは作業者があるテキストの各文について要約を作成するうえで重要か否か二値的に判断して分類したものとする☆☆．

また，本稿では入力テキストに対して任意の長さ（要約率）☆☆☆でその一部を出力するアルゴリズムを要

† 大学共同利用機関国文学研究資料館
National Institute of Japanese Literature
†† 奈良先端科学技術大学院大学
Nara Institute of Science and Technology

☆ 重要文抽出タスクとは，テキスト自動要約の 1 つの研究分野で，文を単位とした抜粋をもって要約とするというアプローチである．問題が明確なので，形式化しやすい．

☆☆ 本稿では「要約」がテキストとしてまとまりを持つか，あるいは修辞的に完成しているかは問題にしない．また，ゆれのある判定データとは，複数の要約を持つデータを意味する．

☆☆☆ 本稿では「要約率」をテキスト自動要約で一般に使われている

約モデルあるいは単にモデルと呼ぶ．たとえば，テキストから適当数の文をランダムに（あるいはテキストの先頭から）選び出す手続きも要約モデルである．本稿では特に人間による判定を教師データとして学習する確率的決定木によるモデル（3 章）と，判定情報に依存しない，クラスタリングに基づくモデル（4 章）とを比較し，両者がどの程度判定データに適合するか検討する．また，考察（6 章）では要約モデルの振舞いをゆれとともにテキスト構造の観点からも分析する．

ところで，一般にゆれの原因としては被験者間の言語能力の違いのほかに（要約対象の）文章の内容に関する被験者の知識，関心の差，など様々な要因が考えられるが，本稿の目的はむろん重要文判定作業におけるゆれの由来を特定することではない．本稿の関心はデータに未知の由来によるゆれが含まれているとしたとき，どのような要約モデルがどのような条件でそのデータをよく表現（予測）できるか明らかにすることにある．

2. シナリオ

前章でも触れたように，本稿ではゆれを含む要約生成を 2 つの対極的な学習パラダイムに基づいてモデル化することを試みる．1 つは，正例となる要約データを利用する教師付き（supervised）アプローチ，もう一方は教師データを用いない教師なし（unsupervised）アプローチである．一般に要約モデルを構成するうえで，どのような条件下でいづれのアプローチが優位にあるかという問題は，重要であるにもかかわらずこれまでの内外のテキスト自動要約研究においてまったく明らかにされてこなかった．本稿の目的はこの点を明らかにしようというものである．

ただ注意すべきは，教師付き，教師なしの学習パラダイムには，ともに多種多様なアルゴリズムが存在し，厳密なパラダイム間の比較を行うことは不可能であるという点である．このため，我々は両パラダイムのリーズナブルなベースラインを用いて比較することにする☆．具体的には，教師付きの学習パラダイムのベースラインとして C4.5 に基づいた決定木，後者のベースラインとしては K-means に基づいたクラスタリングを検討する☆☆．

「圧縮率」（compression rate）の意味で使う．たとえば，要約率 10%の要約とは，テキストの 10%相当の長さのテキストを指す．長さの単位は文字，文，文節などありうるが，本稿では特に文を単位とする．

☆ ここでリーズナブルというのは，不当に弱くないという程度の意である．

☆☆ 自動要約の研究では，Carbonell ら[4] の提案する MMR（Maximally Marginal Relevance）と呼ばれる考え方が広く採用されている．これは，query-based summarization（特定の検索要求（query）にフォーカスした文章の要約）を考えるとき，検索要求に該当する互いに似た文を出力するよりはむしろ互いにできるだけ（maximally に）異なる文を出力したほうが要約として好ましいするとする立場である．MMR の定式化は，いろいろな方法が考えられるが，本稿のクラスタリングベースの要約手法もその 1 つ，あるいは一般化と考えられる．ほかに特異値分解（Singular Value Decompostion）を利用した方法も提案されている[8]．一方，MMR とは独立にクラスタリングを用いた要約研究としては Salton[24]，Radev[23] がある．

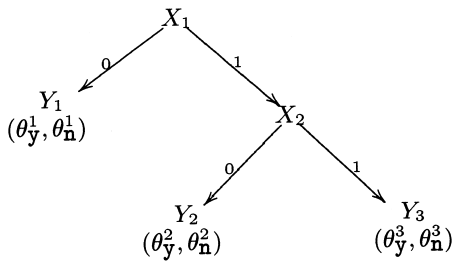

図 1 確率的決定木
Fig. 1 Probabilistic decision tree.

いずれにせよ，ベースラインの構成いかんにかかわらず仮に後者が優位である場合，教師データが不要になるので，要約生成のコスト削減につながることは，付記しておくべきあろう．

3. 確率的決定木に基づく要約モデル

以下では決定木を用いた要約モデルについて説明する．特に本稿では可変長の要約を考えるので，決定木を拡張して，任意の文についてそれが重要文と判定される確率を出力するようにする．確率に基づき文をランキングすることによって，任意の要約率の要約を生成できるからである．以下では確率を出力するための 1 つの手法として確率的決定木を導入する．

例として，2 つのクラスを持つ決定木（**図 1**）を考える．図の中で，X は内部ノード，Y はリーフ，また，結線上の 0，1 は内部ノードの属性値とする．なお，便宜的に内部ノードをその分割属性（splitting attribute）で示すことにする．したがって，X_2 は X_2 を分割属性としたノードを表す．クラスは `y(es)` と `n(o)` とする．さらに，各リーフにはそれぞれのクラスに分類される確率のベクトル $(\theta^i_{\mathtt{y}}, \theta^i_{\mathtt{n}})$ が付随すると考える．

このとき，確率的決定木は，任意のインスタンス（データ）のクラスを出力する代わりにそれぞれのクラスに分類される確率を出力する．したがって，$<X_1=1, X_2=0>$ であるようなインスタンスがあったとき `yes`，`no` それぞれに分類される確率は $\theta^2_{\mathtt{y}}$，$\theta^2_{\mathtt{n}}$ となる．

本稿ではこの分類確率を文のランク付け，任意の要

表 1 文終了形態のエンコーディング
Table 1 Sentence ending forms.

コード（値）	意味
1	非過去（判定詞以外の動詞・形容動詞）
2	過去（判定詞を含む動詞・形容動詞）
3	判定詞（非過去）
4	名詞
5	記号（括弧，ダッシュなど．句読点は除く）
6	終助詞
0	上記以外

約率の要約の生成に利用する．また，決定木としてはC4.5 をベースにし，文単位の確率的分類を行う．以下に，決定木で用いる属性を説明する☆．

LocSen： テキスト中の文の位置を表し，以下で定義する．

$$\frac{Ord(X)}{Ord(LastSentence)}$$

ここで，‘$Ord(X)$’ は，先頭文を 0 とする文 X の順序数．‘LastSentence’ はテキストの最終文．したがって，LocSen のとる値は 0 から 1.

LocPar： 文 X が出現するパラグラフの位置．定義は，LocSen と同様に，

$$\frac{Ord(Par(X))}{Ord(LastParagraph)}$$

で与える．ただし，パラグラフ単位．

LocWithinPar： 文 X のパラグラフ内での相対的位置．先頭位置は 0.

$$\frac{Ord(X) - Ord(ParInitSen)}{Ord(ParLastSen) - Ord(ParInitSen)}$$

ここで，‘ParInitSen’ はパラグラフの開始文，‘ParInitSen’ は終了文.

LenText： テキストの長さ（文字数）.

LenSen： 文 X の長さ（文字数）.

EndCue： 文 X の終止形態を表す．体言止めか，終助詞をともなっているか，動詞終了ならば，過去/非過去か，などの弁別をする．詳しくは，**表 1** を参照されたい.

なお，クラスは ‘Select’ と ‘Don’t Select’ の 2 種類を用いる．これは，文を要約文として選択する，しないの区別に対応する．また以下では，前者を positive（正例），後者を negative（負例）と呼ぶ場合もある．

分類確率を式 (1) で定義する．ただし，本稿では，‘Select’ クラスのみに注目するので，‘Don’t Select’ クラスへの分類確率は無視する．

$$P(Select \mid u, \mathrm{DT}) = \alpha\left(\frac{\text{the number of “Select” sentences at } t(u)}{\text{the number of total instances at } t(u)}\right) \quad (1)$$

式 (1) は，文が決定木 DT のもとで要約文として選択される確率を表す．ここで，u は任意の文，$t(u)$ は決定木 DT において，u が到達したノード．α は適当なスムージング関数とする（本稿では，ラプラス法を用いる）．したがって，要約率 $x\%$ の要約を生成するためには，テキスト中の文を式 (1) に基づいてランキングし，上位 $x\%$ の文を出力すればよい.

さらに本稿では記述長最小原理（MDL）を用いた決定木の最適化を行い，決定木自体の分類精度の向上をはかる（技術的な詳細については付録 A.1 の「MDL による決定木の最適化」の項目を参照されたい）.

4. クラスタリングに基づく要約モデル

次にクラスタリングによる要約の方法について説明する．なお，本章で導入する手法は，基本的に Nomoto ら[18)~20)] 提案の要約手法に基づく.

クラスタリングによる要約では，以下のステップで要約を生成する.

(1) **Find-Diversity**：テキストから多様なトピック・クラスタを構成する☆☆.

(2) **Reduce-Redundancy**：各トピック・クラスタから代表的な文を見つける.

(3) **Generate-Summary**：各クラスタの代表文を適当な順序で並べ出力する.

本稿では，要約文の認識のみを問題にするので，文の出力順序は，ランダムとする．以下では，(1) と (2) のステップについて詳しく見ていく.

4.1 Find-Diversity

Find-Diversity は K-means を MDL で拡張したクラスタリング・アルゴリズムで基本的に X-means 法と同じである[22)]．ここでは，特に X-means と区別す

☆ 一般に位置情報が有用であることが過去の研究で確認されているので，位置情報を中心にした属性構成になっている[1),6),11),27)]．そのほかの属性に関しては，予備実験である程度有効と判断されたものを用いた．手がかり語（EndCue）については，佐久間[29)] を参考にした.

☆☆ トピック・クラスタとは文（ベクトル）を要素とするクラスタ．距離は cosine similarity を用いる．また，あるテキスト T の文 S について，T に現れる記号を除く単語を t_j とするとき $(1 \le j \le N)$，S の文ベクトルを $(w_s(t_1) \ldots w_s(t_N))$ として与える．N は T に現れる記号を除く異なり単語（type）の総数，ここで，$w_s(t)$ を以下で定義する.

$$w_s(t) = \begin{cases} \text{tf.idf}(t) & \text{if } t \in S \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

tf.idf の定義は式 (3) による．なお，ここで単語とは，形態素解析器 ChaSen によって認定された単語を指す[16)]．

るため X^M-means と呼ぶ．

K-means は，データを K 個の互いに素なクラスタに分割する頑健なクラスタリング手法であるが，ユーザが K をあらかじめ指定しなければならない．現実的には求めるクラスタ数が未知の場合が一般的なので，つねに最適なクラスタ数を選択できるとは限らない．X-means のアプローチでは，この問題を解決するため，Bayes Information Criterion（BIC）というモデル選択基準を導入し，最適な分割数を推定できるようにした．X^M-means は基本的に BIC を MDL で置き換えたものと考えてよい．

また，K-means のもう 1 つの問題として，クラスタリングの精度が初期点の選び方に強く依存する，という点が指摘されている[3]．このため，本稿では，200 個，ランダムに初期点を生成し，そのうち最も歪み（distortion）の少ないクラスタリングに至るものを選ぶようにした☆．

X^M-means は，クラスタの歪みを最小にするような開始点を 2 つ生成して，通常の K-means によるクラスタリングを行い，MDL 基準に基づき，再帰的にクラスタを分割していく．また，同時にデータ空間全体をサーチし，つねにデータ記述を最小にするようなクラスタポイントを選択するようにしている．ここで，データ空間とは，1 つのテキストにおける属性表現された文の全体である．

X^M-means を実装したクラスタリング・アルゴリズムを**表 2** に示す．大まかな振舞いとしては，ユーザが指定した K を満たすように，データ空間を K-means で二分していく．ただし，各分割ステップで，MDL を基準に最適な分割点をデータ空間全体を見渡して決定していく．

クラスタの記述長は，決定木の場合と同様，式 (6)（付録 A.1）で定義される☆☆．

表 2 X^M-means アルゴリズム．c_0 はデータの全空間，L は記述長．c_i^j は，クラスタ（j は親クラスタのインデックス）．2-means は K-means（$K = 2$）

Table 2 The X^M-means Algorithm. c_0 here stands for the entire data space, L for the description length. c_i^j indicates a cluster originating from a cluster indexed with j. 2-means indicates K-means with $K = 2$.

X^M-means(c_0,$K_{\max}$)
begin
$C = \phi$
$(c_1^0, c_2^0) = 2\text{-}means(c_0)$
$C = C \cup \{c_1^0, c_2^0\}$
$k = 2$
while $k < K_{\max}$ and there is no convergence
　begin
　$S = \{c : c \in C, L(2\text{-}means(c)) < L(c)\}$
　if S is not empty **then**
　　c_{best}=arg $\min_{c \in S} L(2\text{-}means(c))$
　　$(c_1^k, c_2^k) = 2\text{-}means(c_{best})$
　　$C = C \backslash \{c_{best}\} \cup \{c_1^k, c_2^k\}$
　　$k = k + 1$
　endif
　end
end

4.2 Reduce-Redundancy

次に要約プロセスの第 2 ステップである Reduce-Redudancy についてみていく．このステップでは，Find-Diversity で生成されたクラスタそれぞれについて，代表的な文を同定することを目標するが，ここでは，tf.idf に基づいた単純な文のランキング・モデルを考えることにする．基本的に Zechner[28] のモデルと同じである．

文ランキングは，クラスタ内の文 s について，次式による重み $W(s)$ に基づき決定する．

☆ S をクラスタの集合としたとき，その歪みの度合いを以下で定義した．

$$D(S) = \sum_{c_i \in S}^{k} V(c_i),$$

ただし，$V(c_i)$ は以下で定義する．ここで，μ_i はクラスタ c_i のセントロイド（centroid）とする．

$$V(c_i) = \frac{1}{\mid c_i \mid} \sum_{x_j \in c_i} ||x_j - \mu_{(i)}||^2.$$

☆☆ MDL におけるデータ記述長は，多変量の確率密度関数を使って定義する[5]．ただし，各変量は，正規分布し，分散は等しいとする．具体的には，あるクラスタ c_i の記述長を以下のように定義する．

$$\hat{l}(c_j) = -\log \prod_{i \in c_j} \hat{P}(x_i)$$

ここで，$\hat{P}(x_i)$ はデータポイント x_i（つまり，文）がクラスタ c_i に生起する確率の最尤推定である．

$$\hat{P}(x_i) = \frac{R_{(i)}}{R} \cdot \frac{1}{\sqrt{2\pi\hat{\sigma}}^U} \exp\left(-\frac{1}{2\hat{\sigma}^2}||x_i - \mu_i||^2\right),$$

ただし，$||\ \cdot\ ||$ はユークリッドノルム（Euclidean norm），$R_i = |c_i|$，R はデータの総数，U は次元数（属性数）を表す．μ_i はクラスタ c_i のセントロイド．$\hat{\sigma}^2$ は分散の最尤推定値で，以下で定義される．

$$\hat{\sigma}^2 = \frac{1}{R - K} \sum_{i}^{K} ||x_i - \mu_i||^2.$$

K はデータ空間内の全クラスタ数，x_i はクラスタ i の要素．$R - K$ は自由度．
また，モデル・パラメータは，クラスタの総数 K，分散 $\hat{\sigma}$，各 U 次元セントロイドと考える．さらに，モデルの記述長は $L(M) = -\log|\mathcal{M}|$ とした．ただし，$|\mathcal{M}| = K_{max}$．K_{max} はユーザが指定するクラスタの最大数．つまり，異なった K を持つクラスタリングが同じ確率で生起すると考える．

$$W(s)=\sum_{x\in s}\underbrace{(1+\log(tf(x)))\cdot\log(N/df(x))}_{\text{tf.idf}} \quad (3)$$

ここで，x は 1 文内の単語を表す．各文はその $W(s)$ に基づきランクが付与される．ランク 1 位の文がクラスタの代表として選択され，Reduce-Redudancy の出力となる．$tf(x)$ は x の文書内頻度，N は文書総数，$df(x)$ は x の文書頻度（x が出現した文書数）を表す☆1,18)~20)．上記式のブレイス部分が一般に tf.idf と呼ばれる．

ただ，予備実験で文の長さによる重みの正規化はランキング精度を劣化させることが明らかになった．このため本実験では正規化処理は行わないこととした．

5. テストデータと実験手続き

要約データは，関東，関西の大学生 112 名に要約文抽出作業を行ってもらい収集した．抽出作業は，1995 年度日本経済新聞 CD-ROM 版の社説，報道，春秋の各ジャンルから 25 編，合計 75 編の記事を読み，各記事について要約するうえで最も重要だと判断される文を 10%選択する，というものであった．1 記事あたりの平均文数は 19.98（17.04（春秋），22.32（社説），17.6（報道））であった（以降，このデータセットを JFD-1995 と呼ぶ）．また，被験者間の判定の一致の度合いをみるため，各記事について，平均で 7 名の人に作業してもらった．

作業結果について κ 値を測定したところ，0.25 であった☆2．これは，過去に報告されている一致率の傾向と合致していると思われる[8),24)]．ただ，なかには，高い一致率を得たという報告もあるが，これは作業条件の特殊性，一致率の計測手法に依存するところが大きいと考えられる☆3,9),15)．いずれにせよ，どのような条件のもとで高い一致率が得られるのか，という問題は現在未解決であり，高い一致を持った判定をシステマティックに収集することは現実的に難しいといえる．

☆1 $tf(x)$，$df(x)$ の定義は文献 18)~20) に準ずる．なお，後述の実験では 1995 年日経新聞 CD-ROM 版から人事情報を除く単語数 250~2000 の記事（14,391 件）を抽出し，N，$df(x)$ の各値を決定した．

☆2 κ はカテゴリ判定の被験者間一致率のメトリック．被験者のペアを単位として算出した一致の度数の割合を期待値で補正したもの[26)]．具体的には次式で算出する．$\frac{P(A)-P(E)}{1-P(E)}$．ここで，$P(A)$ はペア単位の一致率，$P(E)$ は一致の期待値．

☆3 高い一致率は，作業条件というより，むしろ計測手法による誤差の可能性が高い．というのは，高い一致率を報告している研究はいずれもメトリックとして Percent Agreement を用いている．このメトリックは重要文判定のような正例が少なく，負例が大多数のようなバイアスのあるデータに対しては，一致率を膨張（inflate）させることが分かっている．

表 3 JFD-1995 の詳細．N はテストデータ中の文数．K はしきい値．$K=n$ は得票数 n 以上の文を正解とする
Table 3 JFD-1995. N represents the number of sentences in JFD-1995. K is an agreement threshold. $K=n$ means that sentences with votes $\geq n$ are marked as positive.

K	N	正例数	負例数	κ
1	1424	707	717	0.253
2	1424	392	1032	0.381
3	1424	236	1188	0.500
4	1424	150	1274	0.600
5	1424	72	1352	0.711

本稿は，特に判定のゆれと要約モデルの精度の関係に注目しているが，様々なレベルのゆれ（一致率）をともなった自然発生的なデータは現実的に入手が困難なため，JFD-1995 から正例（重要文）と見なすための条件を操作し，様々な一致率を持つデータを人工的に合成することにした．

具体的には，複数判定者らによる重要度判定を持つ各文に対して，正例と見なすためのしきい値（K）をもうけ，しきい値に満たない文は負例とした（ただし，記事中の文がいずれもしきい値を満たさない場合，記事そのものをテストデータから除外した）．実際には重要度判定は,「重要だと思う・思わない」の二値判定であるので，しきい値の対象は「思う」と判定した作業者の数になる．各しきい値と正・負の分布の詳細は**表 3** に載せた．

また実験では，MDL による最適化を施した決定木による要約システム（DT/MDL）と，X^M-means を用いたクラスタリングによる要約システム（DBS/X^M）とリードベースの要約システム（LEAD）の比較を行う．LEAD は，ベースラインシステムで記事の先頭から要約率相当数の文を要約として出力する．なお，以下では，文脈から判断して特に誤解が生じないと思われる場合，簡単のため DT/MDL を DT，DBS/X^M を DBS と略記することにする．

実験は，それぞれの手法を用いてテストデータの記事を 10%から 50%まで，10%刻みの要約率で要約し，その中に含まれる正例数で評価した☆4．クラスタリングを用いた手法では，要約率に応じてクラスタの数を増減し，生成された各クラスタから最上位の文を収集し，要約とした．一方，決定木を用いた手法では，記事に現れた文を正例と分類される確率でランク付けし，要約率に相当する数だけ上位から文を抽出した．

☆4 本稿における，正解要約の考え方については付録 A.3 を参照のこと．

表 4 MDL の効果（二値分類タスク）．数字はエラー率．ベースラインではテストデータをすべて負（大多数クラス）として分類．訓練，テストデータいずれも $K = 1$
Table 4 The effects of MDL on a two-class decision tree. Figures in the table indicate error rates. The baseline here labels everything as negative. The testing and training data are constructed with $K = 1$.

	C4.5 (CF=100%)	C4.5(CF=25%)	MDL 最適化	ベースライン
$K1$	0.411	0.396	0.369	0.49

表 5 決定木のジャンル別分類精度（マイクロ適合率）
Table 5 Micro-precision.

K	春秋	社説	報道
K1	0.488	0.516	0.480
K2	0.265	0.276	0.284
K3	0.157	0.167	0.173
K4	0.101	0.098	0.118
K5	0.049	0.038	0.068

実験結果に移る前に，DT/MDL の（確率的でない）二値分類器として精度を確認しておくことにした．結果は**表 4**，**表 5** に載せた．

MDL の効果は，表 4 をみると，確認できる．MDL ベースの決定木は明らかに C4.5（CF=25%）の精度を上回っている．表 5 では，K は前述のしきい値．たとえば，$K = 1$ のとき 1 名以上の被験者が重要と判断した文をすべて正例とする．テスト法は，10-foldCross Validation（CV）．ただし，ここでは単純に正負をまとめたエラー率を測るだけではなく，「マイクロ適合率」を導入し正例に注目した精度も測定した（式 (4)）．二値分類レベルでは，ジャンルによる精度差はあまりないことが分かる．

$$\text{マイクロ適合率} = \frac{\text{真の正例数}}{\text{正例と分類した文数}} \qquad (4)$$

6. 結果と考察

実験の全般的な結果は，**表 6**，**表 7**，**表 8**，**表 9**，**表 10** のとおりである．各表は，正例判定のしきい値（$1 \le K \le 5$）別に見たときの結果であるが，数値は要約モデルの記事単位精度，F-measure の 10-fold CV による平均を表している．F-measure は，$F = \frac{2PR}{P+R}$ とした（いわゆる，F1）．P は precision，R は recall☆．たとえば，表 6 は，$K1$ であるときの（少なくとも被験者 1 名が要約に必要と判定した文をすべて正例扱いにしたときの）各要約モデルの精度を示している．また，表中の α はシステムが用いる要約率を表す．K によって分割される正例，負例の分布については表 3 を参照されたい．ゆれの度合いとモデルの精度との関係を分かりやすくするため，各 K に対応する一致率を付記した．また，データの特性とそれぞれのシステムの振舞いとの関連をみるため，システムの精度はジャンル別にまとめることとした．

以下では，便宜的に得票が 1〜3 のものをマージナルな（重要）文，4 以上のものを主要な（重要）文と呼ぶことにする．

まず，全体的な結果について概観する．春秋をみると，DBS が特に高い性能を示していることが分かる．ただ，K のレベルによらず，ほぼ一貫して $\alpha = 10$，20%で LEAD に劣っている．一方，DT は K のレベルが上がるにつれて次第に性能が回復しており，$K4$ では他システムに対する優位傾向が特に顕著になる．対して，LEAD は，$\alpha = 10$，20%では精度が高いが，α の増加とともにパフォーマンスが落ちる．

社説において際立った特徴としては，α の増加とともに，DBS，DT のパフォーマンスがほぼ一貫して向上することである．LEAD は K のレベルに関係なく α が 10，20%のとき最も良く，α が増加すると，精度が落ちてくる．対して，DT は，LEAD，DBS と比較すると，精度が著しく悪いことが分かる．

報道では，LEAD が一貫して高い性能を示している．ただ，DT は，要約率が 10，20%の場合，DBS を上回る．この傾向は K のレベルが高くになるにつれてさらに顕著になる．DBS は他システムに比べて劣る．

それでは，なぜそれぞれのシステムがこのような振る舞い方をするのであろうか．答えを探るため，まず，重要文の分布を調べてみることにした．

図 2 および**図 3** は，各 K，各ジャンルについて，テキストを 10 のブロックに等分したとき，各ブロックにおける得票率の分布を示したものである．ここで，

☆ テキスト T の正例の全体を C_T とし，システムが正しく検出できた重要文の数を m とするとそのリコールは通常 $\frac{m}{|C_T|}$ であるが，要約率が α のときシステムが獲得できる最大の正例数は $\alpha \cdot |C_T|$ であるから，このままではリコールを少なく見積もってしまう．この点を補正するため，評価では正例の最大数 D を以下のように要約率に依存したかたちで与える．

$$D = \begin{cases} \alpha \cdot |T| & \text{if } \alpha \cdot |T| < |C_T| \ (\alpha > 0) \\ |C_T| & \text{otherwise} \end{cases} \qquad (5)$$

表 6 $K1$ における DT，DBS，LEAD のジャンル別精度
Table 6 Genre-wise performance of DT, DBS, and LEAD on $K1$.

	春秋			社説			報道		
α	DT	DBS	LEAD	DT	DBS	LEAD	DT	DBS	LEAD
10%	0.573	0.694	0.667	0.507	0.637	0.773	0.707	0.707	0.787
20%	0.510	0.711	0.630	0.520	0.650	0.720	0.600	0.687	0.740
30%	0.491	0.594	0.569	0.547	0.617	0.600	0.571	0.636	0.678
40%	0.514	0.583	0.551	0.541	0.611	0.557	0.548	0.631	0.598
50%	0.541	0.596	0.584	0.543	0.628	0.554	0.546	0.631	0.591

表 7 $K2$ における DT，DBS，LEAD のジャンル別精度
Table 7 Genre-wise performance of DT, DBS, and LEAD on $K2$.

	春秋			社説			報道		
α	DT	DBS	LEAD	DT	DBS	LEAD	DT	DBS	LEAD
10%	0.376	0.392	0.416	0.227	0.383	0.520	0.560	0.493	0.627
20%	0.355	0.406	0.408	0.273	0.442	0.473	0.456	0.475	0.560
30%	0.349	0.372	0.358	0.280	0.417	0.368	0.428	0.485	0.517
40%	0.410	0.415	0.323	0.320	0.412	0.328	0.429	0.506	0.484
50%	0.408	0.413	0.345	0.348	0.422	0.350	0.425	0.494	0.472

表 8 $K3$ における DT，DBS，LEAD のジャンル別精度
Table 8 Genre-wise performance of DT, DBS, and LEAD on $K3$.

	春秋			社説			報道		
α	DT	DBS	LEAD	DT	DBS	LEAD	DT	DBS	LEAD
10%	0.208	0.229	0.253	0.168	0.213	0.341	0.444	0.385	0.500
20%	0.212	0.272	0.261	0.203	0.296	0.312	0.374	0.365	0.473
30%	0.237	0.282	0.235	0.203	0.317	0.273	0.350	0.424	0.444
40%	0.304	0.307	0.206	0.213	0.300	0.233	0.342	0.407	0.386
50%	0.302	0.295	0.220	0.240	0.302	0.238	0.333	0.367	0.351

表 9 $K4$ における DT，DBS，LEAD のジャンル別精度
Table 9 Genre-wise performance of DT, DBS, and LEAD on $K4$.

	春秋			社説			報道		
α	DT	DBS	LEAD	DT	DBS	LEAD	DT	DBS	LEAD
10%	0.191	0.142	0.194	0.126	0.130	0.295	0.412	0.357	0.460
20%	0.211	0.181	0.207	0.129	0.211	0.269	0.340	0.336	0.441
30%	0.193	0.215	0.186	0.118	0.217	0.214	0.284	0.366	0.390
40%	0.234	0.218	0.156	0.145	0.196	0.181	0.264	0.317	0.313
50%	0.225	0.203	0.152	0.188	0.195	0.166	0.266	0.290	0.268

表 10 $K5$ における DT，DBS，LEAD のジャンル別精度
Table 10 Genre-wise performance of DT, DBS, and LEAD on $K5$.

	春秋			社説			報道		
α	DT	DBS	LEAD	DT	DBS	LEAD	DT	DBS	LEAD
10%	0.144	0.147	0.175	0.093	0.113	0.273	0.402	0.271	0.441
20%	0.163	0.158	0.167	0.098	0.159	0.222	0.328	0.260	0.392
30%	0.136	0.153	0.136	0.071	0.134	0.162	0.276	0.268	0.311
40%	0.188	0.154	0.107	0.070	0.117	0.141	0.228	0.228	0.246
50%	0.165	0.136	0.099	0.093	0.137	0.128	0.212	0.213	0.204

得票率とは，各ブロックに属する文の総得票を総投票数で割ったものである☆．したがって，あるブロックの得票率が高ければ，そのブロックの文に多くの投票があることを意味する．

☆ 図 2，3 では，各ジャンルごとに，すべてのテキストの得票をブロックごとに累積したものをブロック別の総得票数としている．また，総投票数は，ブロックごとの総得票数の総和で与える．

図 2 では左から春秋，社説，報道，また，最上段が $K1$，下段が $K2$ を表す．図 3 は最上段から下に向かって，$K3$，$K4$，$K5$ を表している．したがって，図 2 の左上隅のパネルは春秋，$K1$ のときの得票率の分布を表している．また，右下隅のパネルは報道で $K2$ のときの分布を示している．

図 2，3 をみるとジャンルに固有の分布パターンが

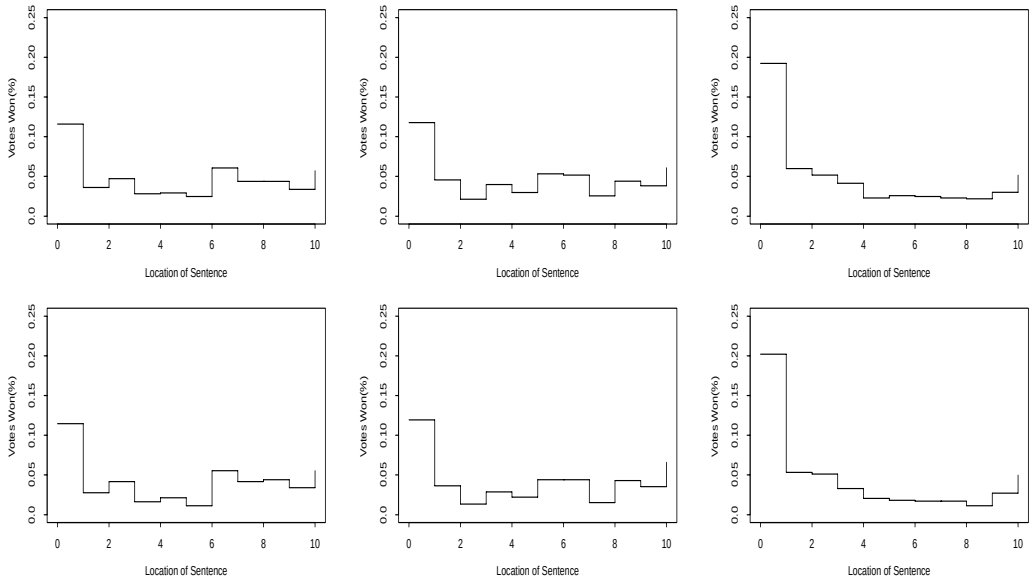

図 2 列は左から春秋，社説，報道の各ドメイン，行は上から $K1$，$K2$ の各データを表す
Fig. 2 Each panel here shows the ratio of votes won by a block of sentences over the total number of votes cast, for various Ks and domains. The rows (from top to bottom) respresent $K1$ and $K2$. The columns respresent various domains ('Shunju,' 'Shasetsu,' and 'Hôdô,' from left to right).

あることが分かる．興味深いのは，どのジャンルでも分布パターンが概して K のレベルにかかわらず一定していることである．春秋では，テキストの冒頭部と結尾部に得票の偏りがみられるが，これはどの K についてもいえることである．一方，社説では，冒頭，中央部，結尾部に偏りが目立つ．K の増加とともに冒頭部，中央部の偏りが目立ってくる．これに対して，報道では，K のレベルによらず，一貫して冒頭部に強い偏りがみられる．

得票率のパターンは一種の文章構造を形成していると考えられるが，重要なことはこれらのパターンがゆれの影響をあまり受けないことである．K が高くても，たとえば，春秋，社説では後半部に得票率の山がみられる．これは，ゆれが少なくても，主要文がテキストの後半でも現れること示している．主要な重要文でも広範囲に分布しうるのである．

それでは，元の問題に立ち戻って，観察された得票率のパターンとそれぞれの要約システムの振舞いの関係を考えていこう．

春秋では，DBS が他システムに比べて優位であった．DT は，K の増加にともなって精度が向上し，LEAD は，α が増加すると精度が劣化することが分かっている．

まず，DBS が良いのはなぜか．これは，春秋における得票率のパターンをみると納得できる．つまり，重要文がテキスト全体にわたって散らばっているからである．K が高くなっても，その傾向は続くから，パフォーマンスはそれほど落ちない．

DT はゆれの影響で K が低いと精度が伸びないが，K が上がるに従って，学習効果が上がり精度が向上する（$K4$ で特に優勢になる）．また，K が低い場合は，冒頭に，特にマージナルな重要文が多く残っているので，LEAD の精度は高い．しかし，K が増加するにつれ，マージナルな文が消えていくと精度が落ちてくる．これは，テキスト後半の重要文を拾えないからである．ただ，α が低いときは冒頭部の重要文を拾うだけでよいから精度は高くなる．

社説もほぼ春秋と同じ説明が可能であるが，DT の精度が著しく低い．これはなんらかの理由で学習が失敗していると考えられるが，原因は今のところよく分かっていない．DBS は，春秋の場合と同じく，α の上昇とともに精度が向上する．これは広範囲に文を拾ったほうが当たる確率が上がるからである．LEAD の振る舞い方は，春秋の場合と同じように説明できる．

報道は，LEAD が他システムを圧倒している．これは，重要文の大多数がテキストの冒頭に集まっているという文章構造上の特性によるものである．DT は DBS に比べて α が低い場合に特に優勢になる．これは学習効果によるものと考えれる．ただ，K が低い場合は，ゆれの影響でパフォーマンスは悪い．DBS の

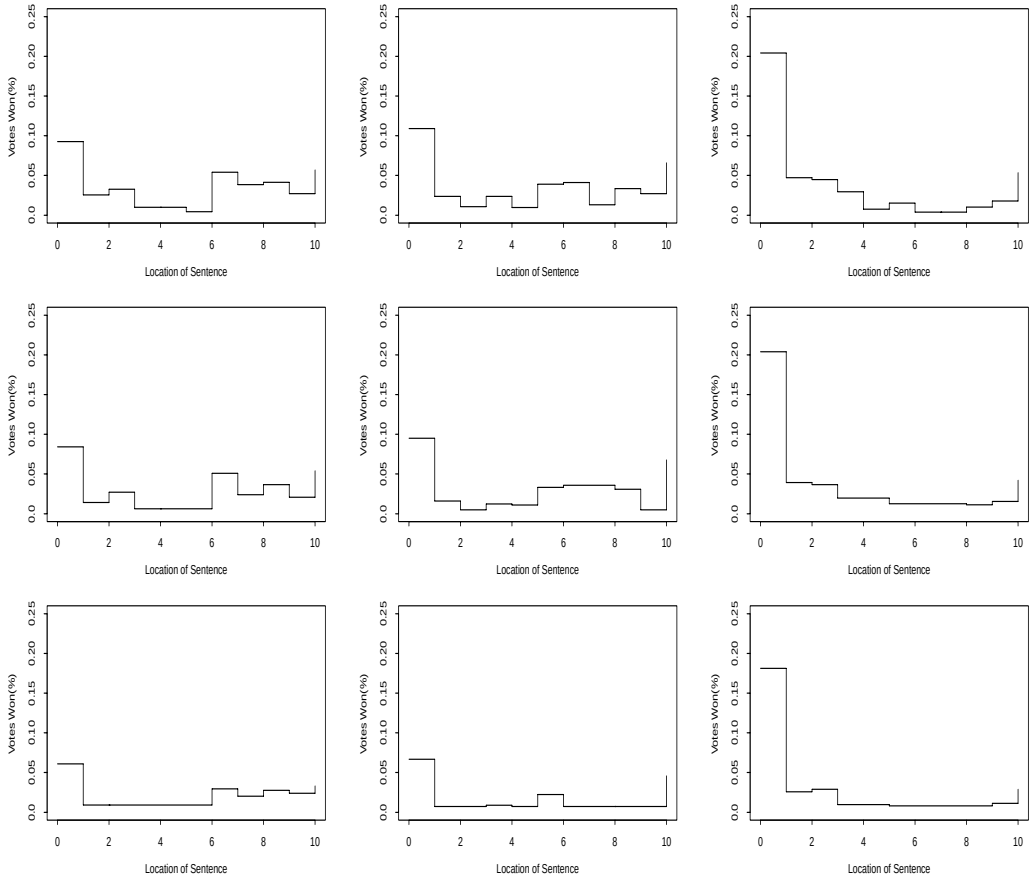

図 3 図 2 の続き．列は左から春秋，社説，報道の各ドメイン，行は上から $K3$，$K4$，$K5$ の各データを表す
Fig. 3 Continued from Fig. 2. We are looking at results for $K3$ (top row), $K4$ (center row) and $K5$ (bottom row).

性能が劣るのは，報道の場合，広範囲に文を拾っても意味がないからである．

以上，結果を簡単にまとめると下のようになると思われる．

- 重要文が広範囲に分散しているかどうかはゆれの大きさと直接関係しない．
- DT はゆれの影響を受けるが，DBS はあまり受けない．特にゆれが少なくなっても DT と同程度の，ときには良い精度がでる．全般的に DBS が DT に比べて優勢である．ゆれが少ないから DT，大きいから DBS という図式は必ずしも成立しないようである．
- α が低い場合，DT が DBS に比べて全般的に優位な傾向を示す．これは学習の効果によるものと考えられる．反対に DBS は位置を無視するため，特定の位置（冒頭部）の重要文を見つけることができない．
- DT，DBS，LEAD のいずれもパフォーマンスが文章構造と深く関わる．特に DT，DBS は重要文が広範囲に分散しているときに優勢になる．

DT の二値分類精度（マイクロ適合）とランキングに基づく要約モデルの精度の関係をみてみると，興味深いことに必ずしも相関しないことが分かる．たとえば，$K1$ では，分類精度は社説が一番良いが，要約モデルでみた場合，報道が一番良い．この違いはいうまでもなく，確率ランキングによるものである．

ところで，Nomoto ら[19] はクラスタリングベースの要約モデル（DBS，本稿提案方式と同一手法）を BMIR-J2 と呼ばれる新聞記事検索課題に適用したところ，LEAD と比べて優位であったと報告している☆．

☆ 具体的には，各新聞記事を機械要約で置き換え，それらを通常の文書のサロゲート（surrogate）と見なして適当な検索エンジンを用い文書検索を行う．評価は，検索要求の要約前の記事への関連性（relevance）の判定に基づいて行う．したがって，当該タスクでは，どれだけ機械要約がもとのテキストと検索要求の関連性を保持しているかというのが問題になる．むろん，同一記事が複数の互いに異なった検索要求に対して関連性を持ちうる．

一般に LEAD では，記事の先頭部分が選択されるため，要約は記事のメイン・トピック中心の単観点の内容になる傾向にあるが，DBS では，先頭に限らず記事の様々な箇所から文が選択されるため，複数観点の要約が構成されやすい．検索課題では，同一記事でも複数の異なった観点で検索対象になることが十分考えられる☆．

本研究において，α の値を上げていくと，特に社説，春秋で DBS が優勢になることが観察されたが，これは，まさに検索課題での DBS の結果と符号する．α が高くなると，より多くの重要文を検出することが必要になるが，春秋，社説の場合，重要文はテキスト冒頭だけではなく末尾にも現れる．このため，DBS が優勢になっていくと考えられる．

このことは，DBS の性質が検索課題と要約課題の 2 つのまったく異なった評価タスクで確認されたことを意味する．

7. むすび

以上，テキスト要約における被験者判定のモデル化という立場から，確率的決定木による要約手法と，クラスタリングに基づく要約手法について，いくつかのモデルを構成して比較検討した☆☆．

今回の研究では大学生 112 名から要約データ収集するという，要約に関するデータ収集としては比較的大きな規模で，テストデータを構築した．

これらのデータに基づく実験を行い，被験者間の一致の度合（K），データのジャンルとそれぞれの要約手法の精度との関連を観察した．実験の結果によると，ゆれは DT の性能に影響を与えるものの，DBS, LEAD に対してはあまり影響がみられないこと分かった．DBS がゆれに影響されないというのは興味深い結果である．

さらに実験は，重要文の分布について，それぞれのジャンルが固有のパターンを持ち，ゆれよりもむしろ，文章構造の違いが要約モデルの性能に大きな影響を与えることを明らかにした．DBS および DT は，重要文がテキストに広く散らばっているとき，優位になるが，重要文が冒頭に極端に偏って現れる報道では，LEAD が圧倒的に優勢になる．

したがって，実験結果は要約モデルを適切に選択するには，ゆれというより文章構造への注意が必要であることを示している．特に教師つきの要約モデルは，SVM [25] を含めた様々な改良の方法がありうるが，重要なのは適用するジャンルを適切に選ぶということである．どのような学習パラダイムにせよ，教師付きの要約モデルを報道に適用しても，効果はあまり期待できない．

ところで，DBS/X^M あるいは決定木による要約手法にはいくつかバリエーションがありうることは容易に想像がつく．たとえば，ランキングモデルとして Zechner モデルの代わりに確率的決定木を用いて，DBS/X^M と決定木を混合するという方法や X^M-means の代わりに，近年注目されている Spectral Clustering [10),17)] などの手法を使うという方法である．また，確率的決定木の強力なバリエーションとして ADTree（Alternating Decision Tree）[7] などの導入も検討に値するだろう．これらの改良，拡張は今後の研究課題としたい．

謝辞 本稿執筆にあたり，有益な助言をいただいた査読者の方々に感謝いたします．

☆ 検索課題では，A 判定，B 判定の記事を正解記事としている．A 判定記事とは，検索要求が主題になっている記事，B 判定記事とは，検索要求が主題からはずれるがそれに関連する内容が含まれている記事である．Nomoto ら [19] の実験では LEAD 法は A 判定記事の検索には強いが，B 判定記事も正解した場合，DBS 法による検索が顕著に優勢という結果が得られている．

☆☆ 2 つの手法で属性をそろえるべきではないかという議論に関しては，付録で本研究の立場を説明しているので参照されたい．

参考文献

1) Aone, C., Gorlinsky, J., Larsen, B. and Okurowski, M.E.: A Trainable Summarizer with Knowledge Acquired from Robust NLP techniques, *Advances in Automatic Text Summarization*, Main, I. and Maybury, M.T. (Eds.), pp.71–80, The MIT Press (1999).
2) Berger, A. and Mittal, V.O.: Query-Relevant Summarization using FAQs, *Proc. 38th Annual Meeting of the Association for Computational Linguistics*, Hong Kong, pp.294–301 (2000).
3) Bradley, P.S. and Fayyad, U.M.: Refining Initial Points for K-Means Clustering, *Proc. 15th International Conference on Machine Learning* (*ICML98*), pp.91–99, Morgan Kaufmann (1998).
4) Carbonell, J. and Goldstein, J.: The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries, *Proc. 21st Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia, pp.335–336 (1998).
5) Duda, R.O., Hart, P.E. and Stork, D.G.: *Pattern Classification*, 2nd edition, John Wiley & Sons, Inc. (2001).
6) Edmundson, H.P.: New Method in Automatic Abstracting, *J. ACM*, Vol.16, No.2, pp.264–285

(1969).
7) Freund, Y. and Mason, L.: The alternating decision tree learning algorithm, *Proc. 16th International Conf. on Machine Learning*, pp.124–133, Morgan Kaufmann, San Francisco, CA (1999).
8) Gong, Y. and Liu, X.: Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis, *Proc. 24th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, New Orleans, ACM-Press (2001).
9) Jing, H., Barzilay, R., McKeown, K. and Elhadad, M.: Summarization Evaluation Methods: Experiments and Analysis, *AAAI Symposium on Intelligent Summarization*, Stanford Univesisty, CA (1998).
10) Kannan, R., Vempala, S. and Vetta, A.: On Clusterings: Good, Bad and Spectral, *Proc. Symposium on Foundations of Computer Science* (2000). Final version to appear in *J. ACM*.
11) Kupiec, J., Pedersen, J. and Chen, F.: A Trainable Document Summarizer, *Proc. 14th Annual International ACM/SIGIR Conference on Research and Developmnet in Information Retrieval*, Seattle, pp.68–73 (1995).
12) Li, H.: A Probabilistic Approach to Lexical Semantic Knowledge Acquistion and Structural Disambiguation, Ph.D. Thesis, University of Tokyo, Tokyo (1998).
13) Lin, C.-Y. and Hovy, E.: Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics, *Proc. HLT-NAACL*, Edmonton, ACL (2003).
14) Mani, I., House, D., Klein, G., Hirshman, L., Obust, L., Firmin, T., Chrzanowski, M. and Sundheim, B.: The TIPSTER SUMMAC Text Summarization Evalutation Final Report, Technical report, MITRE, Virginia, USA (1998).
15) Marcu, D.: The automated construction of large-scale corpora for summarization research, *Proc. 22nd International ACM/SIGIR Conference on Research and Development in Informational Retrieval*, Berkeley, pp.137–144 (1999).
16) Matsumoto, Y., Kitauchi, A., Yamashita, T. and Hirano, Y.: Japanese Morphological Analysis System ChaSen version 2.0 Manual, Technical report, NAIST, Ikoma (1999). NAIST-IS-TR99008.
17) Ng, A.Y., Jordan, M.I. and Weiss, Y.: On spectral clustering: Analysis and an algorithm, *Proc. Advances in Neural Information Processing Systems 14* (*NIPS 14*) (2002).
18) Nomoto, T. and Matsumoto, Y.: An Experimental Comparison of Supervised and Unsupervised Approaches to Text Summarization, *Proc.2001 IEEE International Conference on Data Mining*, San Jose, pp.630–632, IEEE Computer Society (2001).
19) Nomoto, T. and Matsumoto, Y.: A New Approach to Unsupervised Text Summarization, *Proc. 24th International ACM/SIGIR Conference on Research and Development in Informational Retrieval*, New Orleans, ACM (2001).
20) Nomoto, T. and Matsumoto, Y.: The diversity-based approach to open-domain text summarization, *Information Processing and Management*, Vol.39, pp.363–389 (2003).
21) Papineni, K., Roukos, S., Ward, T. and ing Zhu, W.: BLEU: a Method for Automatic Evaluation of Machine Translation, *Proc. 40th Annual Meeting of the Association for Computational Linguistics*, pp.311–318 (2002).
22) Pelleg, D. and Moore, A.: X-means: Extending K-means with Efficient Estimation of the Number of Clusters, *Proc. 17th International Conference on Machine Learning* (*ICML2000*), Stanford University, pp.727–734, Morgan Kaufmann (2000).
23) Radev, D.R., Jing, H. and Budzikowska, M.: Centroid-based summarization of multiple documents: sentence extration, utility-based evalutation, and user studies, *Proc. ACL/NAAL Workshop on Summarization*, Seattle, WA. (2000).
24) Salton, G., Singhal, A., Mitra, M. and Buckley, C.: Automatic Text Structuring and Summarization, *Advances in Automatic Text Summarization*, Mani, I. and Maybury, M.T. (Eds.), pp.342–355, The MIT Press (1999). Reprint.
25) Schölkopf, B., Burges, C.J.C. and Smola, A.J. (Eds.): *Advances in Kernel Methods: Support Vector Learning*, The MIT Press (1998).
26) Siegel, S. and Castellan, N.J.: *Nonparametric Statistics for the Behavioral Sciences*, 2nd edition, McGraw-Hill (1988).
27) Watanabe, H.: A Method for Abstracting Newspaper Articles by Using Surface Clues, *Proc. 16th International Conference on Computational Linguistics*, Copenhagen, Denmark, Vol.2, pp.974–979 (1996).
28) Zechner, K.: Fast Generation of Abstracts from General Domain Text Corpora by Extracting Relevant Sentences, *Proc. 16th International Conference on Computational Linguistics*, Copenhagen, pp.986–989 (1996).

29) 佐久間まゆみ（編）：文章構造と要約文の諸相，くろしお出版 (1989).
30) 奥村，原口，望月：決定木学習を用いたテキスト自動要約手法に関するいくつかの考察，情報処理学会第 59 回全国大会講演集，第 2 分冊，5N-2, pp.393–394, 情報処理学会 (1999).
31) 市川 孝：文章論概説，教育出版，東京 (1990).

付 録

A.1 MDL による決定木の最適化

MDL を用いて決定木を最適化するには，決定木の各部分木をモデルと見なし，そのモデルのもとでのデータの記述長を求め，それが最小になるような部分木を求めればよい．

一般にデータ $x^n = x_1, \ldots, x_n$ が与えられたとき，あるモデル M のもとでので記述長を以下の式で定義する．

$$L(x^n : M) = L(x^n \mid M, \theta) + L(\theta \mid M) + L(M) \tag{6}$$

ここで，$L(x^n \mid M, \theta)$ はデータの記述長，$L(\theta \mid M)$ はモデル・パラメータの記述長，$L(M)$ はモデルの記述長を表す．ただし，θ は，モデル・パラメータのベクトルとする．データ記述長は，一般にデータの最尤推定量をもとに与えられる．また，パラメータの記述量は，自由なパラメータ数を k，データの総数を N とすると，一般に $\frac{k \log N}{2}$ で近似できることが知られているので[12]，u を頂点を u' とする部分木，x^n を u' に到達したデータとすると，データ，パラメータの記述長としては以下の量を考えればよい．

$$I(u) = -\log P(x^n) + \frac{k}{2} \log N \tag{7}$$

ここで，$I(u) = L(x^n|u, \theta) + L(\theta|u)$．これを多少書きかえると，

$$I(u) = -\sum_{j=1}^{K} F_j \log \hat{P}_j + \frac{k}{2} \log N \tag{8}$$

となる．ここで，K は，データ $x_1 \ldots x_n$ に出現するクラス数を表す．また，$\hat{P}_j$ はクラス j の出現確率の最尤推定値，F_j はデータ中にクラス j が出現した頻度を表す．

また，モデル（部分木）の記述長は式 (9) で与える．ただし，P_0 は，決定木におけるリーフノードの生起確率，P_1 は非終端ノードの生起確率を表す．ノードが非終端の場合，さらに分割属性（splitting attribute）のエンコードが必要になるのでそれに要する符号の長さ $l(A)$ が加算される．ただし，A は分割属性．

表 11 MDL による決定木の最適化
Table 11 Pruning with MDL.

```
MDL-Prune(u)
begin
if u is a leaf then
  set L(u) = − log P0 + I(u)
  return L(u)
else
  L(u) = Σ_{v∈D(u)} MDL-Prune(v)
  where D(u) is a set of daughter nodes of u.
  if − log P0 + I(u) ≤ − log P1 + l(A) + L(u) then
    remove every v ∈ D(u)
  endif
  return min{− log P0 + I(u), − log P1 + l(A) + L(u)}
endif
end
```

$$l(u) = \begin{cases} -\log P_0 & \text{if } u \text{ is a leaf} \\ -\log P_1 + l(A) & \text{otherwise} \end{cases} \tag{9}$$

さらに，u' が非終端の場合，子ノードの記述長も考慮するので，その記述長は，

$$-\log P_1 + l(A) + \sum_{v \in D(u')} L(v),$$

となる．ただし，$D(u')$ はノード u' の子ノードの集合とする．

したがって，MDL による決定木の最適化は，決定木をルート・ノードからリーフに向って各中間ノードについて記述量を計算していき，その中で最小記述量の部分木を発見することに帰着できる．具体的には，各中間ノードについてそれをリーフと見なした場合の記述量と非終端と見なした場合の記述量を考え，小さい方を選択していくと，最終的に記述量が最小となった部分木が得られる（具体的なアルゴリズムは，**表 11** 参照）．

A.2 属性の統一と要約手法の比較について

本稿の目的は，2 つの学習アルゴリズム，たとえば，C4.5 と K-means の比較をすることではなく，それらを 利用した2 つの要約手法の性能を比較することである．それぞれの手法は，異なった観点から要約問題にアプローチしており，アーキテクチャーもそれぞれの観点を反映した構成になっている．

たとえば，DBS では，要約は文章中の多様なトピックをできるだけ包含したものでなければならない，という観点が背景にある．この観点は Carbonell[4] の MMR に由来するものである．この意味で DBS は MMR の一形態といえる．

これに対して，決定木による手法では（以下，DT），人間による要約が正しい要約であり，それをモデル化

すべきだという立場に立つ[1),11)].

したがって，属性もそれぞれの立場にとって適当だと思われるものを選択している．たとえば，DBS では位置情報を用いていないが，これはその出現位置に関係なく文を文章全体から拾うことが，トピックのカバー率の向上に貢献すると考えるからである．

一方，決定木においては「キーワード頻度」を用いていない．理由は DBS で使っているような数千次元のキーワード集合をそのまま属性として導入しても，一般的に各キーワードの頻度が高くないため学習効果が期待できないと考えるからである．

また，経験的にもキーワード情報が要約に有効であるとは必ずしもいえない．奥村ら[30)]は，JFD-1995 を用いて，位置情報，文章の長さ，文の tf.idf，接続詞の種類など多様な属性を導入した C4.5 で要約生成の実験を行ったところ，最良値（F1）で 0.439 を得たと報告している．我々の DT 手法で，奥村らの実験条件にできるだけ近い状態で実験したところ，F1=0.464 を得た（奥村らの実験条件の詳細が不明なため，3 名以上が一致した正の判定を正例と見なした．ちなみに本実験における正例の総数は 236，奥村らの正例数は 202 である）．比較実験の結果からは，JFD-1995 に関する限り，tf.idf などのキーワード情報は要約の性能に大きく寄与しないと予想される．

A.3 正解要約について

以下では,「正解要約」について我々の考え方を説明しておく．要約研究では，ある一定の要約率のもとで特定の作業者の作った要約を正解とし評価するのが一般的である．したがって，異なった要約率での精度を測るには，それに応じた正解要約が必要となる．そのため，データ収集のためのコストも増える．

また，要約者個人のバイアスを排除するためには，複数の要約者による複数の要約に対して評価するのが望ましいが，そのとき正解要約をどう定義すべきか自明ではない．残念ながら，今のところ研究者の間でもコンセンサスはない．

たとえば，複数要約の共通部分をとって，それを正解要約しても共通部分は非常に小さくなることが経験的に知られているので，所定の要約率に満たなくなる．一方，複数要約をゆれも含めてすべて正しいとした場合，単純に和をとって正解要約としても，所定の要約率を超過してしまう可能性がある．

最近は BLEU を使った複数要約の評価手法も報告されているが[13),21)]，まだ一般に受け入れられているとはいい難い．すなわち，現時点では要約が複数存在するとした場合，評価上，正解要約をどう定義すべきか結論は出ていないといえる．

このような状況のもと，我々は「正解要約」という概念を別の観点から探ることにした．

まず，その前に今回の人間の判定データ収集作業について多少説明を補足しておく．今回の作業では，判定者に，10%の要約に適した文を選択せよ，という指示ではなく，このテキストについては 3 つ，別のテキストには 2 つ，という具合に，具体的な数をあげて，その数だけ選択せよ，という指示を与えた．むろん，作業者にはその数がテキストの 10%に相当することは知らせていない．また，10%という水準は，作業者がある程度確信を持って文を選択するように企図して設けた水準であり，それ自体特に意味はない．

以下では本編に合わせて，特に判定データ収集時の要約率を γ，モデル評価時の要約率を α，また，重要文の中でも得票数が多い文を主要な重要文，得票数が少ない文をマージナルな重要文と呼ぶ．

ここで正解要約の定義に戻ると，本稿では正解要約を得票数が K 以上の文をすべて重要文とし，それらの中の任意の集合と定義する．特に，α% の正解要約とは正例の中からランダムに選ばれた α%の文の集合と考える☆．たとえば，$\gamma = 10\%$ のとき，$\alpha = 20\%$ の正解要約とは，要約率 10%で収集された要約から合成された 20%の要約と解釈する．

むろん，$\gamma = 20\%$ の場合に比べて $\gamma = 10\%$ の正解要約は数が少なくなるが，ある仮定のもとでは $\gamma = 20\%$ の正解要約として解釈できる（脚注参照）☆☆．

☆ 要約システムは 1 つのテキストについて 1 つの要約を出力すればよいから，正解要約をすべてリコールする必要はない．本稿では，要約率 β% のシステム要約の評価は，生成された要約の中の正例をサブセットとして含むような β% の正解要約を 1 つ適当に定め，その下で行った．いうまでもないが，システム要約の精度は正解要約の選び方によらない．

☆☆ 本稿では，$\gamma = 10\%$，$\alpha = 20\%$ 時の正解要約の集合は，$\gamma = 20\%$，$\alpha = 20\%$ 時の正解要約の集合のサブセットと考える．これは以下の仮定に基づく．

(A) $\gamma \geq 10\%$ の正例集合は $\gamma = 10\%$ の正例集合を部分として含む．

したがって，(A) のもとでは，任意の要約率 $r \geq 10\%$ について，$\alpha = r$，$\gamma = 10\%$ の要約集合 $S_1(r)$ は，$\alpha = r$，$\gamma = r$ の要約集合 $S_2(r)$ に必ず含まれることに注意されたい．また，このとき，$S_1(r)$ を $\gamma = r$ のもとで収集された $\alpha = r$ の正解要約の 1 つのサンプルと見なすことができる．

理由は以下のとおり．まず，(A) をより一般的にとらえ，任意のテキスト T において，$r > u$ であるような要約率 r，u について，$\gamma = u$ のときの正例の集合を $\Gamma(u)$，$\gamma = r$ のときの正例集合を $\Gamma(r)$ とし，$\Gamma(u) \subset \Gamma(r)$ と仮定する．いま，$\Gamma(u)$，$\Gamma(r)$ から $\alpha = r$ であるような要約の集合 $S_r(u)$，$S_r(r)$ をつくることを考える．ただし，$S_r(u) = \{s \subset \Gamma(u) : |s| = r \cdot |T|\}$，$S_r(r) = \{s \subset \Gamma(r) : |s| = r \cdot |T|\}$ とおく．すると，$\Gamma(u) \subset \Gamma(r)$ の仮定より，$S_r(u) \subset S_r(r)$ となる．つまり，$S_r(u)$ を $\gamma = r$ のもとで収集された $\alpha = r$ の正解要約のサ

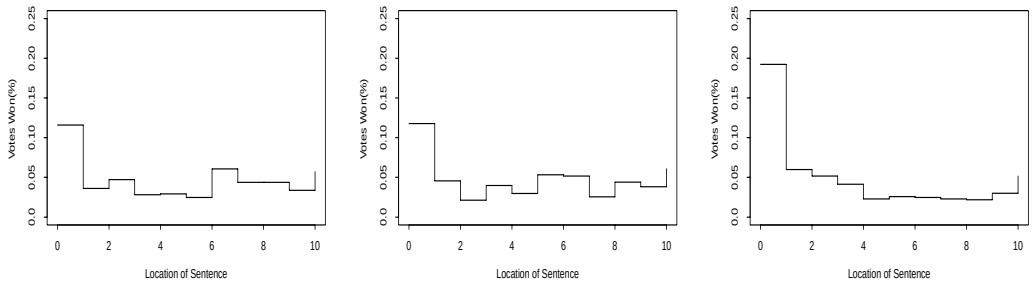

図 4 X は文の位置（%）．Y は得票率．左から，春秋，社説，報道
Fig. 4 X indicates the location of sentence, Y indicates the ratio of votes won over the total number of votes cast. The columns represent domains, 'Shunju,' 'Shasetsu,' and 'Hôdô,' from left to right.

表 12 $K1$ における，$\gamma = 5\%$（A）と $\gamma = 10\%$（B）時の各モデルの F1 値
Table 12 Performance in F1 of the models on $K1$, at $\gamma = 5\%$ (A) and $\gamma = 10\%$ (B).

	春秋			社説			報道		
正例数	DT	DBS	LEAD	DT	DBS	LEAD	DT	DBS	LEAD
A	0.302	0.346	0.332	0.254	0.345	0.379	0.315	0.361	0.380
B	0.573	0.694	0.667	0.507	0.637	0.773	0.707	0.707	0.787

特に，$\gamma = 10\%$ の正解要約の集合には，$\gamma = 20\%$，あるいはそれ以外で選択される可能性のある重要文は，部分的に含まれているものと考える．

では，一般に，$\alpha = 20\%$，$\gamma = 10\%$ のときのモデルのパフォーマンスが，$\gamma > 10\%$ のとき，どのようになるか考察してみたい．

ここで $K1$ における得票率の分布パターン（図 2）を，いま一度，振り返ってみる．なお，利便のため，再度図を掲載しておく（**図 4**）．本編でも述べたとおり，図はテキストを等幅な 10 のブロックに分割したとき，各ブロックに現れた重要文にどのくらいの作業者の投票があったかを，総投票数との比（得票率）でみたものである．

図をながめると，ジャンルごとに固有の得票率の分布パターンがあることが分かる．春秋では，冒頭部と結尾部分，社説は，冒頭部分，中間部分，結尾部分，報道は冒頭と，それぞれ特徴的なピークを持っている．

国語学ではテキストにおける主題の出現パターンをいくつかの文章構成上の型に分類している[29),31)]．たとえば，主題を冒頭に持つ文章は頭括型と呼ばれる．その逆が尾括型．ほかに中括，両括，散括などの型が提案されている．

ここで，重要文を主題（文）と考えてみると，文章構成上，春秋が両括型，社説が散括型，報道が頭括型と言語直感に比較的合った分類が可能であることから，得票率のパターンは，ジャンル固有の主題構造を反映しているという見方ができる．

さらに，言語学的には報道であれば冒頭にピークあり，春秋であれば冒頭と結尾にピークがあると考えるのは不自然ではない．もしこのような直感が正しいとするならば，図の得票率のパターンはすでに安定しており，γ をさらに大きくとってもそれほど変化しないのではないかと予想される．

本稿 6 章の実験結果から，要約モデルの性能は概して重要文の分布パターンに依存しているといえるから，分布パターンが $\gamma = 10\%$ で安定しているという立場に立てば，$\gamma > 10\%$ の場合でもモデル間の優劣はあまり変化しないのではないかという予想が成り立つ．

たとえば，報道で $\alpha = 20\%$，$\gamma = 20\%$ の場合，やはり LEAD が DBS に比べて優勢になることが予想される．なぜなら，DBS が LEAD に勝るためには，$\gamma = 20\%$ における報道の得票率の分布パターンが春秋型に近づく必要があるが，これは考えにくいからである（反対に，γ の増加により分布パターンが春秋型から報道型へ移行するというのも，現実的に考えにくい）．

ただ，一般に DT と DBS の性能が重要文数が増えると，どうなるかは，ただちに明らかではない．おおまかにいえば，DT については，K が低い場合，ノイズ（ゆれ）の影響で DT が劣勢になり，K の上昇とともに精度が向上していくとみるのは妥当だろう．一方，DBS については，K が低いとき優勢，K が高いとき劣勢になることは十分想像される．

参考のため，$K1$ における正例数をランダムに半分

ンプルと見なすことができる．

に圧縮し，要約率を 10%にして実験を行った．結果は**表 12** のとおりである．表中の A は圧縮後の精度を，B は圧縮前の精度を示している．つまり，この実験では γ が 5%のデータを使って 10%要約の評価していることになる．精度の絶対値は変化しているが，モデル間のランクは保存されていることに注意されたい．

(平成 15 年 7 月 16 日受付)
(平成 16 年 1 月 6 日採録)

野本 忠司（正会員）

慶應義塾大学文学部卒業．上智大学大学院外国語学研究科言語学専攻（修士）修了.（株）日立製作所基礎研究所を経て，現在，大学共同利用機関国文学研究資料館助教授．専門は，自然言語処理．認知科学会，ACL，ACM，IEEE Computer Society 各会員．

松本 裕治（正会員）

1955 年生．1977 年京都大学工学部情報工学科卒業．1979 年同大学大学院工学研究科修士課程情報工学専攻修了．同年電子技術総合研究所入所．1984 年～1985 年英国インペリアルカレッジ客員研究員．1985 年～1987 年（財）新世代コンピュータ技術開発機構に出向．京都大学助教授を経て，1993 年より奈良先端科学技術大学院大学教授，現在に至る．工学博士．専門は自然言語処理．人工知能学会，日本ソフトウェア科学会，言語処理学会，認知科学会，AAAI，ACL，ACM 各会員．