コーパスからの『A のB』型慣用表現の自動抽出

執筆者：梅田 真太郎

指導教官：新納 浩幸

平成9年 3月 1日

目次

Chapter 1

序論

1.1 概要

本論文は、自然言語処理に含まれる研究である「コーパスからの『A の B』型慣用表現の自動抽出」について述べる。ここでコーパスとは大量の文書のことであり、本研究ではこれに日経新聞を使用した。また『A の B』型の表現とは、名詞『A』『B』及びこれらを結ぶ格助詞『の』を組み合わた単語列であり、『A の B』型慣用表現とは『A の B』型の表現の形をとる慣用表現のことである。この『A の B』型慣用表現は、単純に『A』または『B』それぞれの意味をあわせたものが『A の B』の意味とはならないという特徴を持つ。

これについて、「火の車」という慣用表現の英訳を一例に取り説明する。「火」は fire 、「車」は car とそれぞれ翻訳が可能である。しかしながら、「火の車」は直訳である car of fire では、その表現自体が所有する意味と異なることは自明である。

- 火 ⇒ fire
- 車 ⇒ car
- 火の車 ⇏ car of fire

このように、慣用表現を訳すにはその表現に対応した訳を予め用意しておく必要がある。高品質の機械翻訳システムを作成するためには、慣用表現を収集、整理しておかなければならない。

しかしながら、慣用表現の収集及び整理にあたって直接人間が判別していては、非常に膨大な時間を要することになる [1]。そのため、この工程を機械化し、自動抽出する必要がある。本研究では、多変量解析手法の一種である判別分析 [4] を用い、コーパスから「火の車」のような『A の B』型慣用表現の自動抽出を試みる。これにより、処理の対象分野に特有の慣用表現も発見することができる [2]。また自動抽出を行うことにより、『A の B』型慣用表現の定義を考慮することもでき、言語学的見地からも有益である [3]。

1.2 位置付け

慣用表現はその表現を構成している単語間の共起性が強いという特徴があるので、単語間の共起性の強さを測ることができれば慣用表現を抽出できる。単語 x と単語 y の共起性の強さの尺度として、Mutual Information(相互情報量)[8] と Dice 係数 [9] があり、従来、これらの尺度を用いて慣用表現の抽出が行われてきた。ただしこれらの抽出法は、抽出の正解率はよいが、広い範囲の慣用表現を取り出すには不向きである。また『連語は頻発する単語列であり、頻発する特性は単語列の絶対頻度より得られる。』という仮

定を形式化した Cost Criteria という手法は、単語列を一つのユニットとしてみなすことで処理の縮小化の大きさを量的に評価する [7]。しかしながら、Cost Criteria は連語の長さと頻度を単位として計算を行うため、本研究のように連語内の単語数が一定であるような表現を抽出することはできない。

Mutual Information や Dice 係数、Cost Criteria 等の従来手法では、基準そのものを人間が定めているため、うまく機能する場合もあるが、正確さに欠けるところが多い。そのため、データから直接学習を行って分析する手法が望ましい。

学習アルゴリズムである C4.5 [6] は、少数の学習データを用いて慣用表現と一般表現の分別を行う基準を生成する。ただし C4.5 の精度は設定した説明変数に大きく依存する。そのため、本研究で用いる説明変数を使った場合、C4.5 の生成する基準は適合条件を非常に細かく区切ってしまうため、言語学的に適したものとは考えられない。

本研究で用いた多変量解析の一手法である線形判別分析は、ある観点から 2 つのグループに分けられる集合に対して、サンプルの持つ特性から学習を行ない、そのサンプルが 2 つのグループのうちどちらに属しているかを判別する学習アルゴリズムであり、また学習から生成される基準は言語学的にも妥当と考えられるものである。

1.3 本論文の構成

本論文は最初に、この分野で一般的に使用されている従来手法について述べる (第 2 章)。その後、本研究で用いた線形判別分析手法の概略及び本研究への適用法について説明を行った後 (第 3 章)、『A の B』型慣用表現の抽出実験 (第 4 章) 及び考察 (第 5 章)、結論 (第 6 章) へと進む。

また、巻末にはプログラムのソースリストや抽出した表現の分析結果等を添付した。

Chapter 2

従来研究

ここでは、従来より使用されてきた手法 (Mutual Information、Dice 係数、Cost Criteria、C4.5) について説明を行う。

2.1 Mutual Information(相互情報量)

Mutual Information (略して MI) は [8]、単語 x と単語 y とがそれぞれに生起する確率と、同時に生起する確率を比較したものである。それゆえ、x と y の結び付きの強さを測る指標となる。抽出によって相互情報量が大きいものは、連語と判定される。

相互情報量は 2 つの単語に対して定義されるので、この単純な方法は単に 2 つの単語列にのみ適用が可能である。そこで、一般化を行うために以下の方法を取る。

1. 基礎となる語彙 V_0 からはじめる。
2. $I(x,y) > Thr$ ならば、V_n に単語列 "x,y" を 1 単語と考えて加える。
3. 上から、新しい語彙 V_{n+1} を決める。
4. 新しい語彙 V_{i+1} を反映させるために総数を調節する。
5. V_{i+1} を基礎として、最初から再び続ける。

MI を $I(x,y)$ とすると、単語 x と単語 y の間の MI は、以下のように定義される。

$$I(x,y) = \log \frac{P(x,y)}{P(x) \cdot P(y)}$$

ここで、$P(x)$ と $P(y)$ は単語の生起確率で、単語の生起数 $f(x)$, $f(y)$ とコーパス中の全単語数 Nから推測が可能である。

$$P(x) = \frac{f(x)}{N}、P(y) = \frac{f(y)}{N}$$

$P(x,y)$ は、x と y の共起の確率で、上式と同様に推測可能である。

$$P(x,y) = \frac{f(x,y)}{N}$$

ここで、$f(x,y)$ は x と y が同時に生起した数である。

2.2 Dice 係数

Dice 係数は [9]、慣用表現の 1 つの特徴と思われる単語間の共起性の強さを測る尺度の 1 つである。単語 x と単語 y の Dice 係数は以下の式で表せる [9] 。

$$dice(x,y) = \frac{2 \cdot f(x,y)}{f(x) + f(y)}$$

ここで、$f(x,y)$ は x と y が同時に生起した数で、$f(x)$, $f(y)$ は単語 x と y のそれぞれの生起数である。

2.3 Cost Criteria

コストクリテリアは [7]、単語列を一つのユニットとしてみなすことで、処理の縮小化の大きさを量的に評価することもので、

- 連語は頻発する単語列である。
- 頻発する特性は単語列の絶対頻度より得られる。

という仮定に基づいている。しかしながら、単純な絶対頻度のアプローチは、働かない。なぜなら、部分列の頻度は常に元の単語列の頻度より高いからである。一例を挙げると、"in spite" は"in spite of" の部分列であるため、"in spite" は"in spite of" より頻繁に現れる。しかしながら、"in spite" の前後関係が与えられると、"in spite" の後に来るのは高確率で "of" になると予想される。従って、"in spite of" は連語であるが、"in spite" は違うことを考慮に入れなければならない。コストクリテリアの概念は、これを形式化したものである。

形式の定義の前に、以下の表記法を定める。

α	$\cdots$	単語列
$\mid \alpha \mid$	$\cdots$	αの長さ
		(α中に含まれる単語の数)
$f(\alpha)$	$\cdots$	コーパス中のαの生起数

また $K(\alpha)$ を α を一ユニットとして扱うことにより生じるコストの縮小と定義する。

$$K(\alpha) = (\mid \alpha \mid -1) \times f(\alpha)$$

$K(\alpha)$ は以下のように解釈する。コーパス中にある単語列 α が存在し、それは $\mid \alpha \mid$ 単語で構成され、$f(\alpha)$ 回生起したものと仮定する。また、一単語の処理コストを 1 とする。同様に、α を一つのユニットとして処理を行う場合、その処理過程を 1 とする。ある単語列が一度に一つの単語として処理されるような場合、その処理コストはその単語列の長さに比例したものであると仮定することは、合理的である。すなわち、α の処理コストは $\mid \alpha \mid$ である。α を一つのユニットとみなすと、処理過程は $\mid \alpha \mid -1$ まで縮小される。α は $f(\alpha)$ 回現れるので、全コスト縮小は $(\mid \alpha \mid -1) \times f(\alpha)$ となると結論付けることが可能である。これは、$K(\alpha)$ の定義である。

しかしながら実際には、単語列は相互にばらばらにしづらく、問題は単純ではない。単語列 α が β の部分列であるとする。(例えば、 $\alpha =$ "in spite" で、 $\beta =$ "in spite of" の場合) そうすると、

$$f(\alpha) \geq f(\beta)$$

を得る。

さらにまた、単語列 α は、$f(\alpha)$ のうち $f(\beta)$ は β としてとらえられる。それゆえ、α の事実上のコスト縮小は以下のように定義される。

$$K(\alpha) = (|\alpha| - 1) \times (f(\alpha) - f(\beta))$$

最後に、以下のようなステップを踏むことによって、コーパスから連語を抽出することが可能となる。

1. コーパス中のそれぞれの α について $K(\alpha)$ を計算する。
2. $K(\alpha)$ の値を使用して単語列 α をランク付けする。
3. 連語の候補として、高ランクの単語列を抽出する。
4. 連語候補のそれぞれの α について $K(\alpha)$ を計算しなおす。

2.4 C4.5

AI 手法である C4.5[6] は、記録された膨大な分類データを調べ、特定の例を一般化することによりモデルを帰納的に作る方法の 1 つで、記録の中にあるパターンを見つけて解析することによって、2 番目の分類モデルを作るコンピュータプログラムである。

- 属性 — 値の記述

  1 つのオブジェクトあるいは事例 (case) に関するすべての情報は、あらかじめ決められた性質あるいは属性 (attributes) によって表現できなくてはならない。それぞれの属性は離散値をとる。しかし、ある事例を表現するために使われた属性のタイプは、別の事例で別なタイプとして扱われてはならない。この制約のため、オブジェクトが変化するような構造を持つような問題領域は扱うことができない。

- まえもって定義されたクラス

  事例が割り当てられるべきカテゴリはまえもって準備されていなくてはならない。これは、機械学習における教師付き (supervised) 学習に対応し、解析によって事例の適当なグループを見つける教師なし学習とは対照的である。

- 離散クラス

  クラスは、事例がそのクラスに属するか属さないかを決めることができるように、はっきりと定義できる必要がある。さらに、事例の数はクラスの数より十分多い必要がある。

- データが十分あること

  帰納的一般化は、データの中から同じようなパターンを見つけることによって行われる。もし、たまたま、はっきりと区別できるパターンを見つけることができなければ、この方法は失敗してしまう。この区別は一般に統計的テストによって行っているので、このテストが意味があるためには十分な事例が必要である。必要となるデータの数は、属性やクラスの数と分類しようとするモデルの複雑さなどの要因から決まる。これらの要因が増えると、信頼性のあるモデルを作るために必要なデータの数は増える。簡単なモデルであれば、小数のデータから作ることができる。しかし、複雑な分類モデルを作るには、一般に数百から数千のトレーニング事例が必要となる。

- “論理的” 分類モデル

  C4.5 は、決定木あるいはプロダクションルールの集合を作ることができるのみである。1 つのクラスは、属性の値に関する記述を集めた 1 つの論理的な表現で表すように制限されている。この制限を満足していない分類モデルの一般的な形の 1 つとして、線形判別 (linear discriminant) がある。線形判別では、属性からの寄与を重み付けしてから加え、閾値と比較する。クラスの記述は論理的というより算術的である。

このプログラムのアルゴリズムを以下に示す。

2.4.1 分割統治法

訓練事例の集合 T から決定木を構成する。クラスを $\{ C_1, C_2, ..., C_k \}$ と表す。T がどのような事例になるかにより、以下の 3 つの可能性が考えられる。

- T は少なくとも 1 つ以上の事例を含み、しかも、そのすべての事例が 1 つのクラス C_j に属する場合:

  この場合は、T に対する決定木は 1 つの葉だけからなり、それにクラス C_j のラベルを付与する。

- T が全く事例を含まない場合:

  この場合も決定木は 1 つの葉からなるが、それに付与すべきクラス名は T 以外の情報に基づいて決めなければならない。例えば、問題領域の知識に基づいて、最も頻繁に現れるクラスを選ぶことができる。C4.5 では、事例が全く与えられなかった葉に付与すべきクラスとして、その親ノードの中で最も頻繁に現れたクラスを用いる。

- T が種々のクラスに属する事例を含む場合:

  この場合は、T を部分集合に分割して、各部分集合ができるだけ単一のクラスに属するように改善する。例えば T 10 というテストが選ばれたとして、それは 1 つの属性に基づいて、相異なる値 $\{ O_1, O_2, ..., O_n \}$ を出力とするとしよう。このとき、テスト結果が O_i となるような T 内の事例の集合を T_i とすれば、T は部分集合 $T_1, T_2, ..., T_n$ に分割される。T に対する決定木は、テスト T 10 を行う決定ノードと各出力値に対応する枝からなり、同様な手順がさらに再帰的に繰り返される。すなわち、その i 番目の枝の先には、訓練事例の部分集合 T_i に基づいて同様な手順で構成された決定木がつながれる。

2.4.2 例

訓練事例の集合の分割は、すべての部分集合が単一のクラスに属するようになるまで繰り返される。このプロセスを説明するために一例を表2.1に示す。この小さな訓練集合では、4種の属性と2つのクラスがある。

これらの事例は単一のクラスに属しているわけではないので、分割統治法によりそれらの分割を試みる。ここで" 天候" のテストを選んだとしよう。その出力値は、" 晴れ" と" 曇り"" 雨" の3通りある。" 曇り" のグループはすべて開催のクラスからなるが、" 晴れ" と" 雨" のグループでは複数のクラスが混在している。そこで、さらに" 晴れ" のグループを湿度が75また" 雨" のグループを強風か否かのテストで分割すれば、その結果得られる各グループはすべて単一のクラスからなるようにできる。こうして最終的に得られた分割の様子と、それに対応する決定木を表2.2に示す。

Table 2.1: 小さな訓練集合

天候	温度 (° F)	湿度 (%)	強風?	クラス
晴れ	75	70	真	開催
晴れ	80	90	真	中止
晴れ	85	85	偽	中止
晴れ	72	95	偽	中止
晴れ	69	70	偽	開催
曇り	72	90	真	開催
曇り	83	78	偽	開催
曇り	64	65	真	開催
曇り	81	75	偽	開催
雨	71	80	真	中止
雨	65	70	真	中止
雨	75	80	偽	開催
雨	68	80	偽	開催
雨	70	96	偽	開催

Table 2.2: 事例の最終的な分割と対応する決定木

事例の分割：

- 天候 = 晴れ：
  - 湿度 ≤ 75：

天候	温度 (° F)	湿度 (%)	強風?	クラス
晴れ	75	70	真	開催
晴れ	69	70	偽	開催

  - 湿度 > 75：

天候	温度 (° F)	湿度 (%)	強風?	クラス
晴れ	80	90	真	中止
晴れ	85	85	偽	中止
晴れ	72	95	偽	中止

- 天候 = 曇り：

天候	温度 (° F)	湿度 (%)	強風?	クラス
曇り	72	90	真	開催
曇り	83	78	偽	開催
曇り	64	65	真	開催
曇り	81	75	偽	開催

- 天候 = 雨：
  - 強風 = 真：

天候	温度 (° F)	湿度 (%)	強風?	クラス
雨	71	80	真	中止
雨	65	70	真	中止

  - 強風 = 偽：

天候	温度 (° F)	湿度 (%)	強風?	クラス
雨	75	80	偽	開催
雨	68	80	偽	開催
雨	70	96	偽	開催

対応する決定木：

- 天候 = 晴れ：
  - 湿度 ≤ 75：開催
  - 湿度 > 75：中止
- 天候 = 曇り：開催
- 天候 = 雨：
  - 強風 = 真：中止
  - 強風 = 偽：開催

2.4.3 テストの評価

各ノードで T を分割するテストをどのように選んでも、それが真の意味での分割であれば、最終的には単一のクラスからなる部分集合への分割が得られる。ここで、真の意味での分割とは、空にならない部分集合 $\{ T_i \}$ が少なくとも 2 つ以上できるような分割のことである。もっとも、そのほとんどすべての部分集合がただ 1 つの訓練事例しか含まないことがあるかもしれない。しかし、決定木を構成する目的は、単にそのような分割を何でもよいから見つけることでなく、問題領域の構造を明らかにしたり予測能力を獲得したりすることにある。そのためには、一つ一つの葉において十分な数の事例が必要であり、したがって、あまり細かく分割しすぎないことが望ましい。理想を言えば、最終的に小さな木が得られるように各段階でテストを選びたい。

利得基準

事例の集合 S に対して、$freq(C_i, S)$ は S の中でクラス C_i に属する事例の数を表す。また、集合 S に含まれる事例数を $|S|$ と表す。

事例の集合 S からランダムに 1 つの事例を選びだし、それがクラス C_j に属していると知らせたとする。このメッセージの確立は、

$$\frac{frec(C_j, S)}{|S|}$$

であり、それが伝える情報量は、

$$-\log_2 \left(\frac{frec(C_j, S)}{|S|} \right) \text{ ビット}$$

となる。このようなクラスの所属関係に関するメッセージの平均情報量を求めるために、S 内での頻度で重み付けしてクラス全体に対する平均を求めると、

$$info(S) = -\sum_{j=1}^{k} \frac{frec(C_j, S)}{|S|} \times \log_2 \left(\frac{frec(C_j, S)}{|S|} \right) \text{ ビット}$$

を得る。この量は集合 S のエントロピーとも呼ばれる。これを訓練事例の集合 T に適用すれば、$info$ (T) は T 内のある 1 つの事例が属するクラスを同定するのに必要な情報量の平均値となる。

さて、テスト X の n 通りの結果に合わせて T が分割された後について、同様な評価を考えよう。このときクラスを同定するのに必要な情報量の期待値は、部分集合上で荷重平均をとって、

$$info_X(T) = \sum_{j=1}^{n} \frac{|T_i|}{|T|} \times info(T_i)$$

となる。これらの差

$$gain(X) = info(T) - info_X(T)$$

は、テスト X で T を分割することによって獲得される情報量を表す。この情報量の利得は、テスト X とクラスとの相互情報量とも呼ばれる。これを最大にするようにテストを選ぶ基準を、利得基準と呼ぶ。

具体例として、表 2.1の訓練集合を再び使用する。ここには、9 事例の" 開催" と 5 事例の" 中止" の 2 つのクラスがあり、

$$info(T) = -\frac{9}{14} \times \log_2 \frac{9}{14} - \frac{5}{14} \times \log_2 \frac{5}{14} = 0.940 \text{ ビット}$$

である (これは、 T 内の 1 事例のクラスを同定するために必要な情報量の期待値を表す)。また、" 天候" を用いて T を 3 つの部分集合に分割した後の結果は、

$$
\begin{aligned}
info_X(T) &= \frac{5}{14} \times (-\frac{2}{5} \times \log_2 \frac{2}{5} - \frac{3}{5} \times \log_2 \frac{3}{5}) \\
&\quad + \frac{4}{14} \times (-\frac{4}{4} \times \log_2 \frac{4}{4} - \frac{0}{4} \times \log_2 \frac{0}{4}) \\
&\quad + \frac{5}{14} \times (-\frac{3}{5} \times \log_2 \frac{3}{5} - \frac{2}{5} \times \log_2 \frac{2}{5}) \\
&= 0.694 \text{ ビット}
\end{aligned}
$$

したがって、このテストによる情報量の利得は、$0.940 - 0.694 = 0.246$ ビットとなる。さて、天候の代わりに強風か否かの属性によって T を分割したとしよう。そうすれば、開催 3 事例と中止 3 事例、開催 6 事例と中止 2 事例からなる 2 つの部分集合が得られたであろう。同様な計算により、

$$
\begin{aligned}
info_X(T) &= \frac{6}{14} \times (-\frac{3}{6} \times \log_2 \frac{3}{6} - \frac{3}{6} \times \log_2 \frac{3}{6}) \\
&\quad + \frac{8}{14} \times (-\frac{6}{8} \times \log_2 \frac{6}{8} - \frac{2}{8} \times \log_2 \frac{2}{8}) \\
&= 0.892 \text{ ビット}
\end{aligned}
$$

で、利得は 0.048 ビットとなり、これは先のテストで得られる利得より少ない。したがって、利得基準は、強風か否かの後者のテストよりも、天候についての前者のテストを選ぶことになる。

利得比基準

利得基準は、多数の値を取るテストを偏重する欠陥を持つ。このことは、患者の身分証明を属性として用いるような仮想的な医療診断タスクを考えてみればわかる。身分証明は患者一人一人を識別するためのものであるから、これを属性として用いて訓練事例集合を分割すれば、1 事例だけからなるたくさんの部分集合が得られることになる。当然のことながら、これらの 1 事例からなる部分集合は単一のクラスに含まれるので、$info_x(T) = 0$ となる。したがって、この属性を用いて訓練事例の集合を分割すれば、情報量の利得は最大になる。しかしながら、クラスの予測能力を獲得しようという観点からは、このような分割は全く無益である。

利得基準に伴うこのような偏重は、多数の値を取ることによって得られた利得部分を調整することによって、矯正することができる。ある事例に関して、それがどのクラスに属するかではなく、そのテスト結果自体を伝えるメッセージの情報量を考える。$info(S)$ の定義からの類推により、分割情報量を

$$
split\ info(X) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \times \log_2 \left(\frac{|T_i|}{|T|} \right)
$$

と定める。これは T を n 個の部分集合へ分割することによって得られる全情報量を表す。一方、情報量利得は、そのうちのクラス分けにかかわる部分の情報量を表す。したがって、利得比

$$
gain\ ratio(X) = \frac{gain(X)}{split\ info(X)}
$$

は、分割によって得られる情報量のうち、有益な部分、すなわち、クラス分類に役立つ部分の割合を表す。ところで、分割が自明な分割に近いときは、分割情報量の値が小さいために利得比の値が不安定になる。そこで、利得比基準では、全テスト中で情報量利得が少なくとも平均以上であるという制約下でこの利得比を最大にするテストを選ぶ。

明らかに、この評価基準の下では、患者の身分証明という属性が高く評価されることはない。前述のように、クラスの数を k とすると上式の分子 (情報量利得) は高々 $\log_2(k)$ となる。一方、訓練事例数を n と

すれば、テスト結果も n 通りに分割されるので、分母は $\log_2(n)$ となる。ここで、訓練事例数 n はクラス数 k よりずっと大きいので、利得比は小さな値となる。

前述の例の議論に戻る。天候に関するテストは、訓練事例を 5 個、4 個、5 個の部分集合に分割する。この分割情報は、

$$-\frac{5}{14}\times\log_2\frac{5}{14}-\frac{4}{14}\times\log_2\frac{4}{14}-\frac{5}{14}\times\log_2\frac{5}{14}$$

すなわち、1.577 ビットと計算される。また、前述したように、利得は 0.246 であるので、利得比は $\frac{0.246}{1.577}=0.156$ となる。

テストの候補

テストの候補の集合が与えられれば、評価基準により各候補を評価して、その中で最良のテストを選び出すことができる。

通常、分類器を構成するシステムでは、テストの形式を定めて、その形式で表されるすべてのテストを調べる。1 つのテストでは 1 つの属性だけを扱うのが普通である。なぜならば、それによって木が理解しやすくなるし、また、複数の属性を一度に扱ったときに起こるような組み合わせ爆発の問題が避けられるからである。

C4.5 は、次の 3 種類のテストを生成する仕組みを持っている。

- 離散的な属性に関する "標準的" なテスト。各属性値をそのまま出力値として、1 つの枝に対応させる。
- 離散的な属性に基づく、より複雑なテスト。属性値をいくつかにグループ分けして出力値を割り当てる。
- 連続値を取る属性 A に対して、閾値 Z との比較により、$A \le Z$ または $A > Z$ に分割するテスト。

これらのテストは、訓練事例を分割する際の利得比に基づいて統一的に評価される。また、自明に近い分割を避けるために、『分割の際、少なくとも 2 つの部分集合 T_i はある最小限の数の事例を含まねばならない』という制約を加えることは有益である。このような制約は、特に訓練集合 T が小さいときに効果的である。

2.4.4 連続属性に関するテスト

まず、考えている属性 A の値によって訓令事例集合 T をソートする。その値は有限個しかないので、それらを大きさの順に並べて $\{v_1, v_2, ..., v_m\}$ と表そう。v_i と v_{i+1} の間にある閾値はどれも同じ効果を持ち、属性 A の値が $\{v_1, v_2, ..., v_i\}$ に含まれるか $\{v_{i+1}, v_{i+2}, ..., v_m\}$ に含まれるかにより事例を分割する。したがって、A に基づく分割法は $m-1$ 通りしかなく、そのすべてを調べることは可能である。

通常、閾値として各区間の中点を選ぶ。したがって、i 番目の閾値は、

$$\frac{v_i + v_{i+1}}{2}$$

となる。

C4.5 は、閾値として中点そのものを選ばずに、訓練事例集合上で A が取り得る値の中でその中点を越えない最大のものを選ぶ。こうすることによって、木や規則で用いられる閾値は、実際にデータ中に現れている "意味ある" 値であると保証できる。

Chapter 3

線形判別分析法による『A の B』型慣用表現の抽出

3.1 多変量解析

多変量解析の役割とは、「わからないこと」を明らかにするために関係式を作る過程で、その関係式に用いる係数を求めることである。多変量解析では「結果」のことを目的変数、「原因」のことを説明変数という。多変量解析には2種類ある。「結果」と「原因」との分析、すなわち目的変数と説明変数との分析を、目的変数のある場合の多変量解析という。それに対し、「原因」についての分析、すなわち説明変数だけの分析を、目的変数のない場合の多変量解析という。

3.1.1 目的変数のある場合の多変量解析

目的変数のある場合の多変量解析は、目的変数と説明変数との関係を調べ、関係式を作成し、その関係式を用いて、次の事柄を明らかにする。

1. 予測
2. 判別
3. 潜在能力
4. 評価

1〜3を行う場合、関係式を作る人と使う人が異なる場合がある。このようなとき、関係式を作った人は、式を使う人に、" どんな人を対象として多変量解析を行ったのか" を正確に伝えなければならない。

これに対し4は、関係式を作ったときの対象者がそのまま評価分析の対象となり、関係式を作った人は、そのままその式を使う人となる。

3.1.2 目的変数のない場合の多変量解析

目的変数のない場合の多変量解析は、説明変数相互の関係を調べ、新しい概念のファクターを導く関係式を作成する。このファクターをものさしとして、変数やサンプル(測定対象物)のポジショニングを行い、変数相互あるいはサンプル相互の関連性や類似性を明らかにする。さらにサンプルの類似性を表す得点を用い、サンプルのグループ化を行う。

目的変数のない場合の多変量解析は、新しいファクターがものさしとなるため、どんなにむずかしくてもきちっとした解釈およびネーミングをしなければならない。目的変数のある場合の多変量解析には、このような苦労はなく、算出された関係式や予測値からすぐに結論が導ける。

このことから、「目的変数のある場合」を論理的な多変量解析といい、「目的変数のない場合」を文学的な多変量解析ということがある。

3.1.3 多変量解析で用いるデータについて

サンプルと変数

多変量解析で使うデータ表は、測定対象物を表側、項目を表頭とするのが一般的である。多変量解析では、表側に用いる測定対象物をサンプル、項目を変数という。

オリジナルデータとサマリーデータ

アンケート調査で収集した個々のサンプルデータをオリジナルデータといい、サンプルデータを集計して1つの表にまとめたものをサマリーデータという。

数量データとカテゴリーデータ

データの形態を整理すると2つに大別される。1つは身長のような数量データで、もう1つは血液型のようなカテゴリーデータである。数量データは、データ間の大小関係を比較したり、演算を行ったときに、意味のある数値となるデータである。カテゴリーデータは、データ間の大小比較や演算をしても無意味で、ここでの数値はたんなる分類の意味しか持たない。

数量データのことを量的データあるいは距離尺度、カテゴリーデータのことを質的データあるいは名義尺度ともいう。

3.1.4 多変量解析の手法の選択

手法を選択するためのポイントは次の 2 つである。

1. 明らかにしたいテーマについて、目的変数のある場合、ない場合のどちらの多変量解析を用いれば良いかを調べる。
2. 適用するデータの形態が、数量データなのかカテゴリーデータなのかを調べる。

この 2 つを調べれば、次の表より、選択する多変量解析の手法が決まる。

Table 3.1: 多変量解析の手法

目的変数の有無	データ形態		解析手法
	目的変数	説明変数	
ある場合	数量データ	数量データ	重回帰分析 正準相関分析
	数量データ	カテゴリーデータ	数量化 1 類
	カテゴリーデータ	数量データ	判別分析
	カテゴリーデータ	カテゴリーデータ	数量化 2 類
ない場合		数量データ	主成分分析 因子分析 数量化 4 類
		カテゴリーデータ	数量化 3 類
		(注)	クラスター分析

(注) クラスター分析は、主成分分析、因子分析、数量化 3 類より求められるサンプルの得点を用い、サンプルのグルーピングを行う手法である。

3.2 線形判別分析

3.2.1 判別分析とはなにか

サンプルが持っているいろいろな特性から、そのサンプルがどの群 (グループ) に属するかを判別する手法を、判別分析 [4](discriminant analysis) という。

サンプルがどの群に所属するかを判別するには、何らかの基準を設けなければならない。この手法における判別の基準は次の 2 つである。

1. 線形判別関数による判別
2. マハラノビスの汎距離による判別

これらの判別基準を用い、分析に用いたサンプルがどの群に属するかを推定し、実際の群との対応を調べることによって、判別の精度を調べることができる。よく使われる精度の目安としては、次の 3 つがある。

1. 判別的中率
2. 相関比
3. 誤判別の確率

線形判別関数による判別

まず、「線形判別関数による判別」とはどのような方法かを説明する。

この方法は、群 (グループ) で表されるカテゴリーデータを目的変数にとり、数量データで表される P 個の諸特性を説明変数にとって、

$$Z = a_0 + a_1 \cdot x_1 + a_2 \cdot x_2 + \cdots + a_p \cdot x_p \tag{3.1}$$

という関係式を作る。この式を線形判別関数式 (linear discriminant function) の式といい、これを用いて判別を行う。

マハラノビスの汎距離による判別

「マハラノビスの汎距離」による判別は、判別関数式を作成することもなく、当該サンプルが各群の重心 (平均) までどれほどの距離があるかを調べ、もっとも近い距離の群に所属すると判定する方法である。ここで用いる距離は、日常われわれが使っているものではなく、マハラノビスの汎距離である。なお、この距離は確率で表せるので、最近では予測等の意思決定にこの方法を使うことが多くなっている。

3.2.2 線形判別関数の式

表 3.2の例題を用いて、線形判別関数式の求め方を考える。

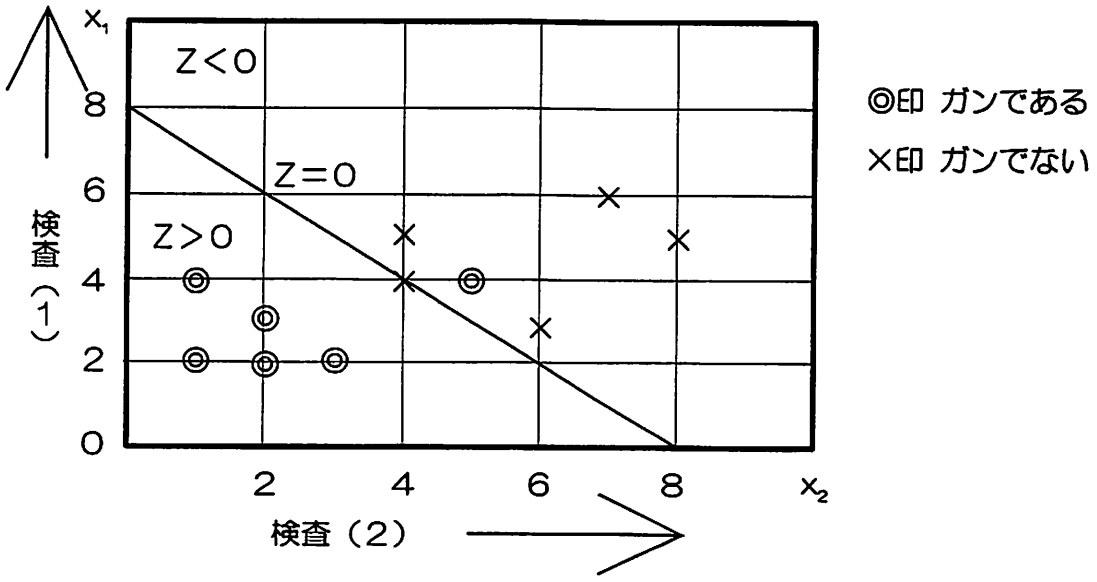
ガンの人 5 人とガンでない人 5 人の計 10 人について、2 つの検査を行ったものとして、これに判別関数式をあてはめてみる。

Table 3.2: 例題

サンプル \ 変数	ガンの有無	検査 (1)	検査 (2)
A	1	3	2
B	1	4	1
C	1	2	2
D	1	2	3
E	1	4	5
F	2	4	4
G	2	5	8
H	2	3	6
I	2	6	7
J	2	5	4
W	?	5	6

(注) ガンの有無　1 = ガンである　2 = ガンでない

ここで縦軸 (x_1) に検査 (1) を、横軸 (x_2) に検査 (2) をとり、点グラフを描いてみる。



この図において、ガンである人 (◎印) とガンでない人 (×印) を 1 本の直線で分けることを考えてみる。直線は何本でも引けるが、2 グループを分けるのに最良な直線が 1 本あるはずである。

この直線を求めたとき、傾きが -1 、x_1 軸切片が 8、すなわち $x_1 = -x_2 + 8$ であったとする。この直線を変形すると

$$0 = -x_1 - x_2 + 8 \tag{3.2}$$

となる。

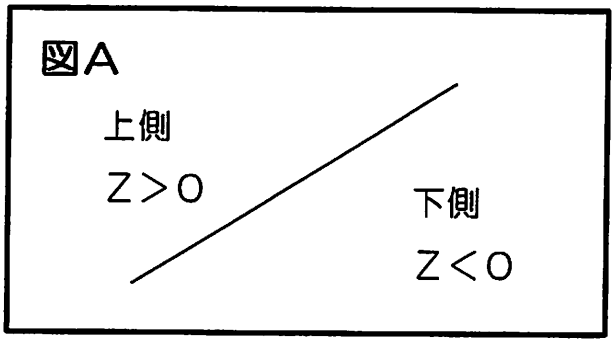
ここで、

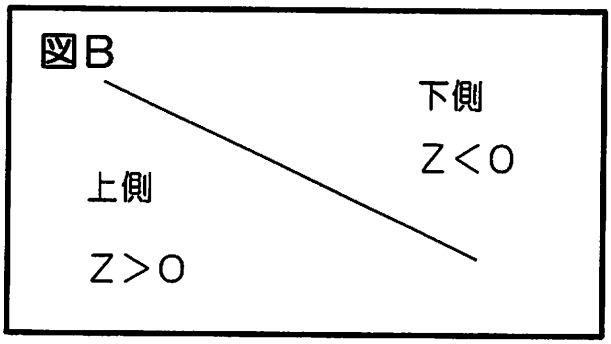
$$Z = -x_1 - x_2 + 8 \tag{3.3}$$

なる関数式を考えたとき、式 3.2 は式 3.3 の Z を 0 に置き換えたものである。

これより式3.3は $\begin{pmatrix} Z=0 & \text{のとき2グループを分ける境界線} \\ Z>0 & \text{ならこの境界線より上側の領域} \\ Z<0 & \text{ならこの境界線より下側の領域} \end{pmatrix}$ を示す関数となる。

ただし、直線の傾きが負 (右下がり) の場合、図上における上側領域と下側領域とが逆になる。





この例の場合、前項のように、グラフは図 B の形をとる。各サンプルの x_1 と x_2 の値を式 3.3 に代入して Z を求めてみる。Z の値が正なら境界線の上側にあり、このサンプルはガンであると判定する。逆に Z が負なら境界線の下側にあり、ガンではないと判定する。$Z = 0$ ならどちらとも言えないと判断する。

このようにして求めた Z の値のことを、**判別得点**という。

判別得点の正負の符号から、各サンプルの理論的な群を推定する。この群を推定群といい、これに対しガンであるかどうかの事実を実績群という。この例における推定群は、判別得点が正のとき「1」、負のとき「2」になる。判別得点が0のときは、推定群を求めることはできない。

Table 3.3: 判別得点

	ガンの有無 (実績群)	判別得点	推定群
A	1	3	1
B	1	3	1
C	1	4	1
D	1	3	1
E	1	-1	2
F	2	0	×
G	2	-5	2
H	2	-1	2
I	2	-5	2
J	2	-1	2

計算例

A における判別得点

$$\begin{aligned} Z &= -x_1 - x_2 + 8 \\ &= -3 - 2 + 8 \\ &= 3 \end{aligned}$$

実際には式 3.3 は、いまの段階ではわからない。そこで x_1 、 x_2 の係数を a_1 、 a_2 、定数項を a_0 として Z を表現すると次式になる。

$$Z = a_1 \cdot x_1 + a_2 \cdot x_2 + a_0 \tag{3.4}$$

この式のことを判別関数式という。

係数を求める公式

ここでの問題は 3.4 式の a_1 、 a_2 、 a_0 を、いかにして求めるかということである。

判別得点より求められた推定群と、実績群 (ガンであるかどうかの事実) とができるだけ一致するように、a_1 、 a_2 、 a_0 の値を決めればよいということになる。

このような a_1 、 a_2 、 a_0 は、次の手順によって求められる。

1. 群ごとのサンプル数及び各変数の平均、分散を、次のように定義する。

	サンプル数	平均		分散	
		変数 No.1	変数 No.2	変数 No.1	変数 No.2
群 1	n_1	$\bar{x}_{1(1)}$	$\bar{x}_{2(1)}$	$S_{11(1)}$	$S_{22(1)}$
群 2	n_2	$\bar{x}_{1(2)}$	$\bar{x}_{2(2)}$	$S_{11(2)}$	$S_{22(2)}$

2. 群ごとに、変数 No.1 と変数 No.2 の共分散を次のように定義する。

群 1	$S_{12(1)}$, $S_{21(1)}$
群 2	$S_{12(2)}$, $S_{21(2)}$

(注) $S_{12(1)}$ と $S_{21(1)}$ の値は同じ

(注) $S_{12(2)}$ と $S_{21(2)}$ の値は同じ

3. 分散，共分散について、2 つの群の加重平均を求め、その値を次のように定義する。

$$
\begin{aligned}
S_{11} &= \frac{(n_1-1)\cdot S_{11(1)} + (n_2-1)\cdot S_{11(2)}}{n_1+n_2-2} \\
S_{22} &= \frac{(n_1-1)\cdot S_{22(1)} + (n_2-1)\cdot S_{22(2)}}{n_1+n_2-2} \\
S_{12} &= \frac{(n_1-1)\cdot S_{12(1)} + (n_2-1)\cdot S_{12(2)}}{n_1+n_2-2} \\
S_{21} &= \frac{(n_1-1)\cdot S_{21(1)} + (n_2-1)\cdot S_{21(2)}}{n_1+n_2-2}
\end{aligned}
$$

加重平均で求められた値を、プール後の分散・共分散という。

a_1 、 a_2 は、次の連立方程式を解けば求められる。

$$
\begin{cases}
a_1\cdot S_{11} + a_2\cdot S_{12} = \bar{x}_{1(1)} - \bar{x}_{1(2)} \\
a_1\cdot S_{21} + a_2\cdot S_{22} = \bar{x}_{2(1)} - \bar{x}_{2(2)}
\end{cases}
\tag{3.5}
$$

定数項 a_0 は、次の式で求められる。

$$
a_0 = -\frac{a_1\cdot(\bar{x}_{1(1)}+\bar{x}_{1(2)}) + a_2\cdot(\bar{x}_{2(1)}+\bar{x}_{2(2)})}{2} \tag{3.6}
$$

計算

ここではこの式を使って係数を求めてみる。

Table 3.4: 例題 (調査結果)

	ガンの有無	検査 (1)	検査 (2)
A	1	3	2
B	1	4	1
C	1	2	2
D	1	2	3
E	1	4	5
F	2	4	4
G	2	5	8
H	2	3	6
I	2	6	7
J	2	5	4

		検査 (1)	検査 (2)
合計	ガンである	15	13
	ガンでない	23	29
平均	ガンである	$\bar{x}_{1(1)}$ 3.0	$\bar{x}_{2(1)}$ 2.6
	ガンでない	$\bar{x}_{1(2)}$ 4.6	$\bar{x}_{2(2)}$ 5.8
分散	ガンである	$S_{11(1)}$ 1.0	$S_{22(1)}$ 2.3
	ガンでない	$S_{11(2)}$ 1.3	$S_{22(2)}$ 3.2

共分散	ガンである	$S_{12(1)} = S_{21(1)} = 0.25$
	ガンでない	$S_{12(2)} = S_{21(2)} = 0.65$

Table 3.5: プール後の分散・共分散

$$S_{11} = \frac{(4 \times 1.0 + 4 \times 1.3)}{8} = 1.15$$
$$S_{22} = \frac{(4 \times 2.3 + 4 \times 3.2)}{8} = 2.75$$
$$S_{12} = \frac{(4 \times 0.25 + 4 \times 0.65)}{8} = 0.45$$
$$S_{21} = \frac{(4 \times 0.25 + 4 \times 0.65)}{8} = 0.45$$

3.5式に代入すると、

$$1.15 \cdot a_1 + 0.45 \cdot a_2 = 3 - 4.6$$
$$0.45 \cdot a_1 + 2.75 \cdot a_2 = 2.6 - 5.8$$

これを解くと、 $a_1 = -1.0$ 、 $a_2 = -1.0$
これを 3.6式に代入して、 $a_0 = 8$ が得られる。
ここで、3.4式に a_1 、 a_2 、 a_0 の値を代入すると、次の判別関数式、

$$Z = -x_1 - x_2 + 8 \tag{3.7}$$

が得られる。

係数の見方

判別関数式では、どの説明変数が大事であるかを、係数の大小比較から明らかにすることはできない。
このことを確かめるために、表3.2の例で検定 (1) のデータを 10 倍して判別関数式を求めてみた。

$$Z = -0.1 \cdot x_1 - x_2 + 8$$

係数の値をみるとわかるように、データを 10 倍すると係数は $\frac{1}{10}$ になる。このように、説明変数のデータ単位によって係数の値は変わるため、係数の大小を比較しても何の意味もない。
「説明変数の大事さ」把握は、判別関数ではマハラノビスの平方距離と F 値を用いる。マハラノビス平方距離が小さいほど、あるいは F 値が大きいほど、その説明変数が大事であることがわかる。

変数選択の方法

変数選択は、次の手順で行う。

1. 目的変数と相関の高い説明変数を選択する。

2. 説明変数相互で高い相関があるとき、どちらかの変数を落とす。

判別関数における目的変数はカテゴリーデータなので、目的変数との相関は相関比を用いる。相関比がいくつ以上のものを選ぶという基準はないが、ここでは 0.5 以上 (単相関は 0.7 以上) とした。これも絶対的な数値でなく、場合によっては基準を下げることもある。
相関比を用いず、群別に説明変数の平均値を求め、平均値を比較し、差があると思われるものを選択することもある。

3.2.3 マハラノビスの汎距離

マハラノビスの汎距離 (Mahalanobis generalized distance) とは、いくつかの点の中から任意の 2 つを選択し、2 点間の距離を計算する際に用いるもので、ユークリッド距離の値にすべての点のばらつきを考慮して算出する。

マハラノビスの汎距離による判別分析

この方法は、個々のサンプルについて、2 つのグループの平均 (重心) までのマハラノビスの汎距離を求め、得られた 2 つの距離の大小関係から、どちらのグループに属するか (近いか) を判定するものである。したがって、どのサンプルについても 2 つのマハラノビスの汎距離を求めることになる。
i 番目サンプルの、両群の重心までのマハラノビスの汎距離を ${D_{i(1)}}^2$ 、 ${D_{i(2)}}^2$ とすると、それぞれは、次式によって求められる。
$\bar{x}_{j(1)}$ ： j 番目変数の第 1 群の平均
$\bar{x}_{j(2)}$ ： j 番目変数の第 2 群の平均

公式

マハラノビスの汎距離

$$D_{i(1)}{}^2 = \sum_{j=1}^{p}\sum_{k=1}^{p}(x_{ij} - \bar{x}_{j(1)}) \cdot S^{jk} \cdot (x_{ik} - \bar{x}_{k(1)}) \tag{3.8}$$

$$D_{i(2)}{}^2 = \sum_{j=1}^{p}\sum_{k=1}^{p}(x_{ij} - \bar{x}_{j(2)}) \cdot S^{jk} \cdot (x_{ik} - \bar{x}_{k(2)})$$

S^{jk} : 次に示す S' の逆行列

$$\begin{cases} S_{(1)} & : \text{第 1 群の偏差平方和・積和行列} \\ S_{(2)} & : \text{第 2 群の偏差平方和・積和行列} \end{cases}$$

$$\begin{cases} S'_{(1)} & : \text{第 1 群の分散・共分散行列 } S'_{(1)} = \dfrac{S_{(1)}}{(n_1)-1} \\ S'_{(2)} & : \text{第 2 群の分散・共分散行列 } S'_{(2)} = \dfrac{S_{(2)}}{(n_2)-1} \end{cases}$$

S' : $S'_{(1)}$ と $S'_{(2)}$ との加重平均

$$S' = \frac{(n_1-1)\cdot S'_{(1)} + (n_2-1)\cdot S'_{(2)}}{(n_1+n_2-2)}$$

(ただし、n_1 、n_2 は両群のサンプル数)

このようにして求められた S' を、プール後の分散・共分散行列という。

3.2.4 個々のサンプルがどの群に属するかを判定

判別得点による判定

線形判別関数によって判別得点を求める。

i 番目サンプルの判別得点を $\hat{Z}_i$ とおくと、

$$\hat{Z}_i = a_1 \cdot x_{i1} + a_2 \cdot x_{i2} + \cdots + a_j \cdot x_{ij} + \cdots + a_p \cdot x_{ip}$$

が成り立つ。このとき、

$\hat{Z}_i$ ＞ なら 1 群に判別される。
$\hat{Z}_i$ ＜ なら 2 群に判別される。
$\hat{Z}_i$ ＝ ならどちらともいえない。

表 3.2の例において、W さんがガンであるかどうかを判定してみる。表 3.2にデータに対する線形判別関数式は 3.7より

$$Z = -x_1 - x_2 + 8$$

これに W さんのデータ $x_1 = 5$, $x_2 = 6$ を代入する。

$$Z = -5 - 6 + 8 = -3 < 0$$

これより、W さんは第 2 群に所属する。すなわち W さんは、「ガンでない」と判定される。

マハラノビスの汎距離による判定

得られた値が
${D_{i(1)}}^2 < {D_{i(2)}}^2$ なら 1 群に判別
${D_{i(1)}}^2 > {D_{i(2)}}^2$ なら 2 群に判別
${D_{i(1)}}^2 = {D_{i(2)}}^2$ なら判別不可能
判別得点による判定と同じように、W さんがガンであるかどうかを判定してみる。
W さんの群 1 および群 2 までのマハラノビスの汎距離を求めると

$$\begin{aligned} {D_{W(1)}}^2 &= (5-3)\times 0.9291\times(5-3)+(5-3)\times(-0.1520)\times(6-2.6) \\ &\quad +(6-2.6)\times(-0.1520)\times(5-3)+(6-2.6)\times 0.3885\times(6-2.6) \\ &= 6.1399 \\ {D_{W(2)}}^2 &= 0.1399 \\ {D_{W(1)}}^2 &> {D_{W(2)}}^2 \end{aligned}$$

これより、W さんは第 2 群に所属する。すなわち W さんは、「ガンでない」と判定される。

3.2.5 判別分析の精度

各々のサンプルがどの群に属するかを推定した結果と、実際に各々のサンプルが所属する群との対応を調べることによって、判別の精度を求めることができる。判別分析でよく使う精度には、次の 3 つがある。

1. 判別的中率
2. 相関比
3. 誤判別の確率

判別的中率

各々のサンプルがどの群に属するかを、判別得点、あるいはマハラノビスの汎距離によって判定する。この判定と、実際に各々のサンプルが所属する群との一致度をみるのが、判別的中率である。

$$\text{判別的中率} = \frac{100\times \text{一致サンプル数}}{\text{分析対象全サンプル}} \tag{3.9}$$

表 3.2の例について判別的中率を求めてみる。

Table 3.6: 例題

サンプル	実績群	判別得点	推定群	マハラノビスの汎距離 (群 1)　(群 2)	判定
A	1	$3 > 0$	1	$0.1399 < 6.1399$	1
B	1	$3 > 0$	1	$2.4101 < 8.4101$	1
C	1	$4 > 0$	1	$0.8865 < 8.8865$	1
D	1	$3 > 0$	1	$1.1128 < 7.1128$	1
E	1	$-1 < 0$	2	$2.4372 > 0.4372$	2
F	2	$0 = 0$	×	$1.2649 = 1.2649$	×
G	2	$-5 < 0$	2	$11.7615 > 1.7615$	2
H	2	$-1 < 0$	2	$4.4912 > 2.4912$	2
I	2	$-5 < 0$	2	$11.8696 > 1.8696$	2
J	2	$-1 < 0$	2	$3.6264 > 1.6264$	2

＊群 ： 1 = ガンである　　2 = ガンでない

実績群と推定群とが一致したサンプルは A、B、C、D、G、H、I、J の 8 名であるので、判別的中率 $= \frac{100 \times 8}{10} = 80$ % となる。

判別得点、マハラノビスの汎距離どちらで行っても、判別的中率の値は同じである。

2 群の判別的中率は 50 % ～100 % の間をとる。

判別的中率が何 % 以上あればよいかという問題は、経験的なものによるが、ここでは次の基準で用いる。

- 90～100% ･･･ 分析の精度が非常によい
- 75～90% ･･･ 分析の精度がややよい
- 50～75% ･･･ 分析の精度がよくない

相関比

相関比は数量データとカテゴリーデータとの相関である。この相関係数を判別分析の精度に用いる。

線形判別関数により求められた判別得点を数量データ、実際に所属する群をカテゴリーデータとして、これらの相関比を求める。

表 3.3の判別得点について相関比を求めてみる。

$$
\begin{aligned}
\text{分散} \quad \sigma^2 &= \frac{\{3^2+3^2+4^2+\cdots+(-1)^2+(-5)^2+(-1)^2\}-\frac{0^2}{10}}{10} \\
&= \frac{96}{10} \\
&= 9.6 \\
\text{級間分散} \quad \sigma_B^2 &= \frac{\{\frac{12^2}{5}+\frac{(-12)^2}{5}-\frac{0^2}{10}\}}{10} \\
&= \frac{57.6}{10} \\
&= 5.76 \\
\text{相関比} \quad \eta^2 &= \frac{\sigma_B^2}{\sigma^2} \\
&= \frac{5.76}{9.6} \\
&= 0.60
\end{aligned}
$$

相関比は0から1の間をとる。判別的中率同様、値がいくつ以上あれば良いという基準はない。ここでは、次のように決める。

- 0.8～1.0　分析精度が非常によい
- 0.5～0.8　分析精度がややよい
- 0.5 未満　分析精度がよくない

誤判別の確率

あるサンプルが1群であるにもかかわらず2群と判定する、2群であるにもかかわらず1群と判定する、といったまちがいのおこる確率を、誤判別の確率という。

2群の重心の距離をマハラノビス平方距離 $\bar{D}_0^2$ で表すと $\bar{D}_0^2$ は次によって求められる。

$$\bar{D}_0^2 = \sum_{j=1} a_j \cdot (\bar{x}_{j(1)} - \bar{x}_{j(2)}) \tag{3.10}$$

a_j は線形判別関数式の係数である。

$\bar{x}_{j(1)}$ は各変数の第1群の平均、$\bar{x}_{j(2)}$ は各変数の第2群の平均である。

$d_0 = \sqrt{D_0^2}$を求め、標準正規分布表で $\frac{d_0}{2} = Z_{(p)}$ なる p を求める。

p を誤判別の確率という。

表3.2の例について、誤判別の確率を求める。

3.2.2係数を求める公式より

$$\bar{x}_{1(1)} = 3.0 \quad \bar{x}_{2(1)} = 2.6$$
$$\bar{x}_{1(2)} = 4.6 \quad \bar{x}_{2(2)} = 5.8$$

3.2.2計算より $a_1 = -1$ 、 $a_2 = -1$

これより

$$\begin{aligned}\bar{D}_0^2 &= a_1 \cdot (\bar{x}_{1(1)} - \bar{x}_{1(2)}) + a_2 \cdot (\bar{x}_{2(1)} - \bar{x}_{2(2)}) \\ &= (-1) \times (3.0 - 4.6) + (-1) \times (2.6 - 5.8) \\ &= 4.8 \\ Z_p &= \frac{\sqrt{D_0^2}}{2} \\ &= 2.19 \div 2 \\ &= 1.095\end{aligned}$$

$Z_{(p)} = 1.095$ となる確率を標準正規分布表より求めると 0.137 となる。
これより誤判別の確率は 13.7% となる。

3.3 『A の B』型慣用表現の抽出

『A の B』型慣用表現の自動抽出のおおまかな流れを以下に示す。

3.3.1 説明変数

本研究では以下のように 2 つの説明変数を設定した [2]。

$$x_1 = \frac{(『A\ の\ B』\ の頻度)}{(『A\ の\ *』\ の頻度)}$$

$$x_2 = \frac{(『A\ の\ B』\ の頻度)}{(『*\ の\ B』\ の頻度)}$$

ここで、x_1 は『A の』という表現の頻度で『A の B』の頻度を割ったもの、つまり『A の』という表現が生起したときの『B』の生起確率である。同様に x_2 は『の B』という表現が生起したときの『A』の生起確率である。

3.3.2 座標化

次に、『A の B』型の表現を説明変数により座標化し、データとしての形を整える。これによりプログラムに入力可能なデータが出来上がる。

3.3.3 学習

座標化された『A の B』型の表現の一部を学習データとして学習させ、線形判別関数を作成する。

3.3.4 分析

作成された線形判別関数を用いてテストデータを分析し、慣用表現の候補の自動抽出を行なう。

Chapter 4

抽出実験

4.1 使用したデータ

本研究で用いたコーパスは、日経新聞'94 年 1〜3 月の 3 カ月分であり、総文数 355,628 文、1 文平均 49.0 文字である。

これを JUMAN (形態素解析システム) [5] で単語切りと品詞付けを行なった (表 4.1 参照)。

```
> juman "私の妹は茨城大学の学生の恋人です。"
```

私	(わたくし)	私	普通名詞
の	(の)	の	名詞接続助詞
妹	(いもうと)	妹	普通名詞
は	(は)	は	副助詞
茨城	(いばらき)	茨城	固有名詞
大学	(だいがく)	大学	普通名詞
の	(の)	の	名詞接続助詞
学生	(がくせい)	学生	普通名詞
の	(の)	の	名詞接続助詞
恋人	(こいびと)	恋人	普通名詞
です	(です)	だ	判定詞
。	(。)	。	句点

Figure 4.1: juman 解析例

その解析結果から、「(複合) 名詞 + の +(複合) 名詞」 の表現を取り出した (表 4.2 参照)。

私	の	妹
茨城大学	の	学生
学生	の	恋人

Figure 4.2: 「A の B」の抽出例

その収集結果から、各『A の B』の表現に対して、以下の形式のリストを作った。

A の B、A、B、『A の B』の頻度 、
『A の＊』の頻度、『＊の B』の頻度、説明変数 x_1 、説明変数 x_2

その一例

所得税の減税　所得税　減税　9　77　163　0.116883　0.055215　0.075000
所得税の税率　所得税　税率　5　77　76　0.064935　0.065789　0.065359
所要の措置　所要　措置　4　16　156　0.250000　0.025641　0.046512
女のユニオン　女　ユニオン　5　89　6　0.056180　0.833333　0.105263
女の子　女　子　42　89　115　0.471910　0.365217　0.411765
女子の採用　女子　採用　5　67　147　0.074627　0.034014　0.046729
女性の割合　女性　割合　4　457　543　0.008753　0.007366　0.008000
女性の活躍　女性　活躍　4　457　51　0.008753　0.078431　0.015748
女性の活用　女性　活用　4　457　142　0.008753　0.028169　0.013356
女性の感性　女性　感性　4　457　18　0.008753　0.222222　0.016842
女性の健康　女性　健康　7　457　27　0.015317　0.259259　0.028926
女性の仕事　女性　仕事　5　457　276　0.010941　0.018116　0.013643
女性の視点　女性　視点　8　457　100　0.017505　0.080000　0.028725
女性の時代　女性　時代　6　457　352　0.013129　0.017045　0.014833
女性の心　女性　心　6　457　154　0.013129　0.038961　0.019640

最終的に取り出したデータは以下のとおりである。

- 『A の B』型の表現の種類数 : 311,966
- 頻度 4 以上の『A の B』型の種類数 : 7,136
- 複合語を除いた『A の B』型の種類数 : 3,172
  これらの表現をテストデータとする. 学習データはこの中の一部を使う.
  その内訳 { 慣用表現 : 27 (内学習データ 15)
  一般表現 : 3,145 (内学習データ 89)
  上記の内訳は [1] により判別した.

4.1.1 慣用表現及びその学習データ

以下はテストデータとして取り出された辞書に登録されている慣用表現 27 種類と、そこに含まれる学習データ 15 種類である。

火の海	車の流れ	生の声	天の声・・・☆
火の気	車の両輪・・・☆	他山の石・・・☆	伝家の宝刀・・・☆
海の幸・・・☆	心の奥	対岸の火事・・・☆	年の瀬・・・☆
気圧の谷・・・☆	心の支え	台風の目・・・☆	悩みの種・・・☆
高根の花・・・☆	心の底	茶の湯	背水の陣・・・☆
座右の銘・・・☆	人の出入り	注目の的	氷山の一角・・・☆
時間の問題	世間の目	長蛇の列・・・☆	

(注) ☆印は学習データとして使用した慣用表現。

4.1.2 一般表現の内の学習データ

以下はテストデータとして取り出された一般表現のうち、学習データとして使用した表現 89 種類である。

ばらの騎士	チームスピリの中止	ビルの屋上	マイカーローの金利
ロシアの選択	安どの表情	一の大きさ	一進一退の状況
一定の歯止め	一定の幅	一般の家庭	一般の消費者
一部の法案	円安の進行	円相場の上昇	家計の負担
過去のデータ	海外の作品	絵の具	各国のコメ
各社のトップ	関係者の話	見直しの対象	減産の影響
現行の中選挙区制	個別の企業	個別の問題	国の文化
国会の場	国債の償還	国連の仲介	今後のスケジュール
最寄りの駅	最後のチェック	昨年の調査	桜の名所
三種の神器	司法の判断	子供の教育	子供の話
氏の発言	私の履歴書	資金の一部	資本金の額
自国の安全	自宅の庭	主婦の姿	受注の内訳
授業の内容	従業員の士気	従来のもの	従来の方法
商店街の活性化	将来の目標	消費者の混乱	状況のなか
状況の下	状況の改善	人々の目	人件費の上昇
政府の決定	絶滅の恐れ	先物の買い戻し	選択の自由
前の年	操業度の低下	相場の基調	損失の穴埋め
大阪の街	第二の人生	中学校の教師	賃上げの原資
党内の結束	東京の中心部	南アフリカの政情	日本の関係者
日本の現在	廃止の方向	売り方の買い戻し	百貨店の売り上げ
評価の対象	米国の需要	米国の要求	貿易の拡大
北朝鮮の出方	名の下	労働者の権利	論議の過程
話し合いのテーブル			

4.2 実験

上記データを用い、線形判別分析、C4.5、MI、Dice 係数それぞれを用いた慣用表現の抽出実験を行った (但し MI,Dice 係数に関しては上位 300 を抽出)。

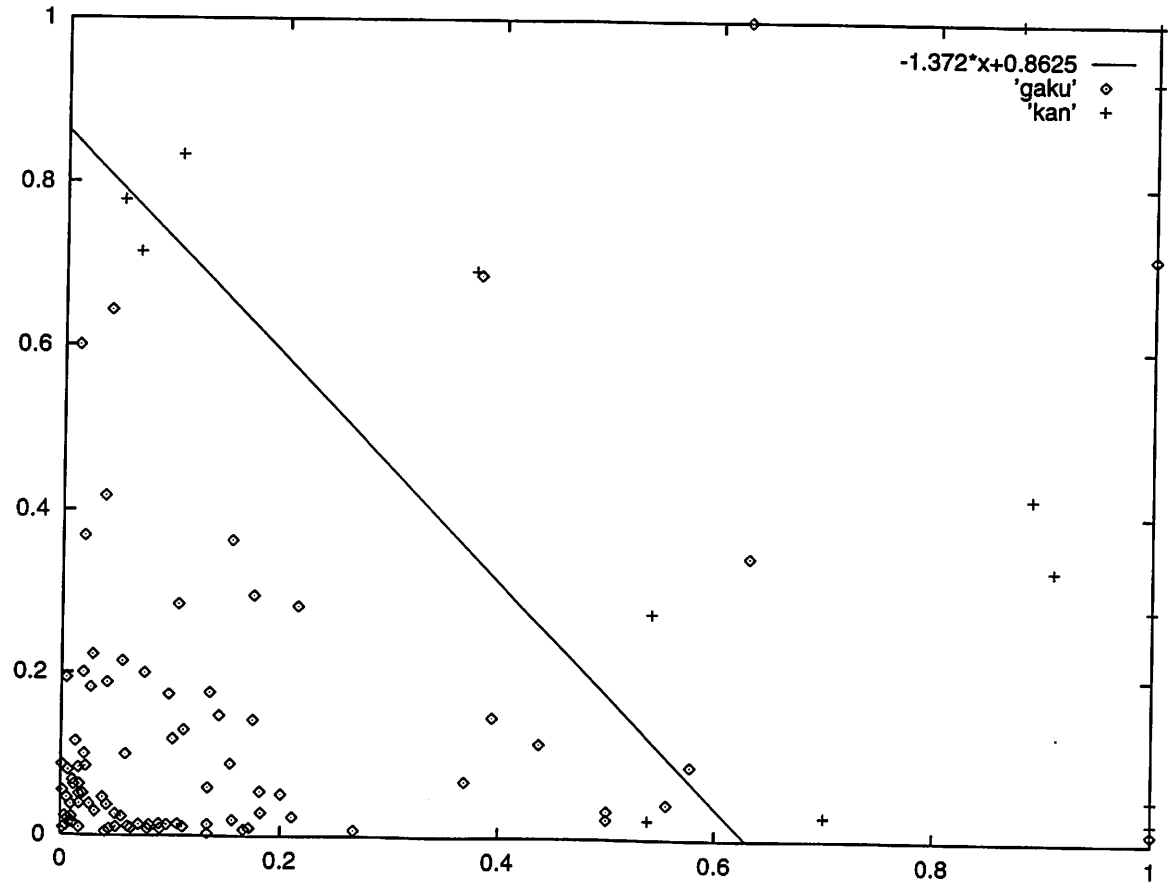
本研究で作成した線形判別分析プログラムの処理過程を以下に示す。

```
サンプル数 15 平均1 0.670038 平均2 0.492944
サンプル数 89 平均1 0.139449 平均2 0.121539

分散1 0.125252 分散2 0.130546 共分散 -0.032824
分散1 0.039686 分散2 0.031909 共分散 0.012189

プール後の分散1 0.052028 分散2 0.046136 共分散 0.005696
求められた係数 a[0] = -5.937688 a[1] = 9.444497 a[2] = 6.884122
```

以下は、実験において求められた線形判別関数式である。

$$Z = 9.444497 \cdot x_1 + 6.884122 \cdot x_2 - 5.937688$$

この式を使い、テストデータを自動的に判別し、慣用表現を判別し、抽出する。学習データの学習による線形判別関数及び学習データを以下に図示する。



4.3 結果

4.3.1 学習データの判別結果

C4.5 で生成されたプログラムを以下に示す。ただしここでのプログラムは決定木の形式で記述する。

```
C4.5 [release 5] decision tree generator Tue Jan 30 18:16:48 2001
----------------------------------------

    Options:
Trees evaluated on unseen cases
File stem <jik>

Read 104 cases (2 attributes) from jik.data

Decision Tree:

attribute#1 > 0.62963 : 1 (11.0/2.0)
attribute#1 <= 0.62963 :
|   attribute#2 > 0.6875 : 1 (5.0/1.0)
|   attribute#2 <= 0.6875 :
|   |   attribute#1 <= 0.5 : 0 (83.0)
|   |   attribute#1 > 0.5 :
|   |   |   attribute#1 <= 0.540541 : 1 (2.0)
|   |   |   attribute#1 > 0.540541 : 0 (3.0)
```

学習データの判別結果を以下に示す。

Table 4.1: 学習データからの抽出結果

	慣用表現 (15)		一般表現 (89)	
	慣用表現と判断	一般表現と判断	慣用表現と判断	一般表現と判断
線形判別分析	12	3	3	86
C 4 . 5	15	0	3	86

上記結果より、学習データの再現性という点においては、C4.5 が線形判別分析を上回っている。

4.3.2 テストデータの判別結果

以下に、テストデータの結果を示す。ここで未登録の慣用表現とあるのは、辞書 [1] には未登録だが主観的判断により慣用表現と認定したものである。

Table 4.2: テストデータからの抽出結果

	抽出数	慣用表現	未登録の慣用表現	誤抽出
線形判別分析	285	13	50	222
C4.5	265	15	45	205
MI(相互情報量)	300	14	46	240
Dice係数	300	12	46	242

実験より、どの手法も大きな違いはないという結果が出た。しかしながら、未登録の慣用表現の抽出においては、本研究で用いた線形判別分析手法が最も優れた値を示している。

線形判別分析による辞書に未登録の慣用表現

以下は線形判別分析手法で取り出された辞書に未登録の慣用表現50種類の一覧である。

おふくろの味	お家の事情	きょうのことば	こどもの国
ばらの騎士	ものの依	もろ刃の剣	オペラ座の怪人
カヤの外	シンドラーのリスト	マーフィーの法則	リオのカーニバル
哀悼の意	一身上の都合	奥の細道	夏子の酒
蚊帳の外	絵の具	胸の内	言葉の端々
五重の塔	交通の便	菜の花	財布のひも
財布のヒモ	三種の神器	死の棘	春分の日
女の子	世の中	成人の日	青天のへきれき
静観の構え	草の根	足並みの乱れ	団塊の世代
塩の浦	地の利	中興の祖	鎮守の森
悩みのタネ	白羽の矢	発祥の地	髪の毛
宝の山	万里の長城	味の素	未婚の母
網の目	腕の見せどころ		

C4.5による辞書に未登録の慣用表現

以下はC4.5によって抽出された辞書に未登録の慣用表現45種類の一覧である。

おふくろの味	お家の事情	きょうのことば	こどもの国
ばらの騎士	ものの依	もろ刃の剣	オペラ座の怪人
カヤの外	シンドラーのリスト	マーフィーの法則	リオのカーニバル
哀悼の意	一身上の都合	夏子の酒	蚊帳の外
胸の内	言葉の端々	五重の塔	国の天然記念物
菜の花	財布のひも	財布のヒモ	三種の神器
死の棘	春分の日	世の中	成人の日
青天のへきれき	静観の構え	足並みの乱れ	団塊の世代
壇の浦	地の利	中興の祖	鎮守の森
白羽の矢	発祥の地	髪の毛	宝の山
万里の長城	味の素	未婚の母	網の目
腕の見せどころ			

MI(相互情報量) による辞書に未登録の慣用表現

以下は MI によって抽出された辞書に未登録の慣用表現 46 種類の一覧である。

いのちの電話	おふくろの味	お家の事情	きょうのことば
ばらの騎士	もろ刃の剣	オペラ座の怪人	カヤの外
シンドラーのリスト	マーフィーの法則	リオのカーニバル	哀悼の意
暗黙の了解	一身上の都合	奥の細道	夏子の酒
火の粉	蚊帳の外	絵の具	胸の内
言葉の端々	五重の塔	交通の便	菜の花
財布のひも	財布のヒモ	三種の神器	死の棘
青天のへきれき	静観の構え	草の根	足並みの乱れ
団塊の世代	壇の浦	地の利	中興の祖
鎮守の森	悩みのタネ	白羽の矢	発祥の地
髪の毛	宝の山	万里の長城	味の素
未婚の母	腕の見せどころ		

Dice 係数による辞書に未登録の慣用表現

以下は Dice 係数によって抽出された辞書に未登録の慣用表現 46 種類の一覧である。

お家の事情	きょうのことば	ばらの騎士	もろ刃の剣
オペラ座の怪人	シンドラーのリスト	マーフィーの法則	リオのカーニバル
哀悼の意	暗黙の了解	一身上の都合	奥の細道
夏子の酒	火の粉	蚊帳の外	絵の具
胸の内	言葉の端々	五重の塔	交通の便
菜の花	財布のひも	財布のヒモ	三種の神器
死の棘	女の子	青天のへきれき	静観の構え
草の根	足並みの乱れ	妥協の産物	団塊の世代
壇の浦	男の子	地の利	中興の祖
鎮守の森	悩みのタネ	白羽の矢	発祥の地
髪の毛	宝の山	万里の長城	味の素
未婚の母	腕の見せどころ		

Chapter 5

考察

コーパスからの慣用表現の抽出では辞書に登録されている慣用表現を取り出すよりも、人間の登録もれのあった慣用表現やそのコーパスで利用されている分野固有の慣用的表現を取り出せる方が有益であり、この点で本手法は他手法よりも優れている。

未抽出の原因

慣用表現であるが抽出されなかったもの (未抽出)14 個について、グラフを書いて分析を行なった。これは未抽出の慣用表現の分析グラフとして巻末に添付されている。結果、未抽出の原因として

- 前 (後) の表現が非常に多いため抽出に失敗した。
- 前 (後) の表現を使用している他の慣用表現があるため数値が低くなり、抽出に失敗した。

等が考えられた。

これを解決するためには、一般表現と判断されたデータから、新たな条件を用いてこれを補うという方法が考えられる。

誤抽出の原因

誤って抽出された一般表現 (誤抽出) についてはランダムに 15 個を抽出し、グラフを書いて分析を行なった。これは誤抽出の表現の分析グラフとして巻末に添付されている。結果、誤抽出の原因として

- 前 (後) の表現が一つしかなかった。
- 頻度が非常に高かった。
- 使用したコーパスが新聞であるため、多少使われた表現にかたよりがあった。

等の原因が考えられた。

本手法の抽出精度は、設定した説明変数に大きく依存するため、適切な説明変数を発見することが重要である。今回用いた前後の単語の共起性という観点以外の新たな観点を導入することによって精度をあげることができると思われる。

Chapter 6

結論

本研究では、多変量解析手法の一種である判別分析を用い、「コーパスからの『A の B』型慣用表現の自動抽出」を試みた。慣用表現はその表現を構成している単語間の共起性が強いという特徴があるので、単語間の共起性の強さを測ることができれば慣用表現を抽出できる。従来、Mutual Information や Dice 係数、Cost Criteria 等の手法ではこの点に着目して、慣用表現を取り出す努力がなされて来た。しかしながらこのような従来手法では、基準そのものを人間が定めているため、うまく機能する場合もあるが、正確さに欠けるところが多い。

また、学習アルゴリズムである C4.5 は、少数の学習データを用いて慣用表現と一般表現の分別を行う基準を生成する。しかしながら C4.5 の生成する基準は適合条件を非常に細かく区切ってしまうため、本研究で用いる説明変数を使用した場合、言語学的に適したものとは考えられない。

本研究で用いた多変量解析の一手法である線形判別分析は、ある観点から 2 つのグループに分けられる集合に対して、サンプルの持つ特性から学習を行ない、そのサンプルが 2 つのグループのうちどちらに属しているかを判別する学習アルゴリズムであり、また学習から生成される基準は言語学的にも妥当と考えられるものである。

慣用表現 15、一般表現 89 の学習データによる学習及びその後のテストデータによる抽出実験の結果、本研究では以下の成果が得られた。

- 従来手法との比較実験を行い、他手法よりも辞書に未登録の慣用表現を多数抽出する点でよい結果が得られた。
- これより本手法は慣用表現収集において有用と考えられる。

Chapter 7

謝辞

本研究の遂行及び論文の作成において多大な御助言及び御指導を賜った新納 浩幸 教官 (茨城大学工学部システム工学科) に深い感謝の意を表します。

また、知識的、精神的、経済的な援助をくださった 水野 孝泰 技官 (茨城大学工学部システム工学科)、尾形 達哉 氏 (茨城大学大学院修士課程)、寺島 成樹 氏 (茨城大学工学部システム工学科 4 回生) にも深く感謝致します。

Appendix A

線形判別分析プログラムソースリスト

```
/*============================================================

線形判別分析を全自動処理するための Program

Written by : Shintaro Umeda (Ver 1.00)

      処理を行うことが可能なデータは、以下の形式のみである。

      問題点 :


  ============================================================*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct{
  char tango[1000];
  char mae[1000];
  char ato[1000];
  int kazu;
  int kazum;
  int kazua;
  float kazumh;
  float kazuah;
  float kazuw;
} retu;

main(){
```

```
int i;
int sample[2];              /*  サ ン プ ル 数  */
float heikin[4];            /*       平 均       */
float bunsan[4];            /*       分 散       */
float kyoubunsan[2];        /*     共 分 散      */
float kajubun[2];           /*    加 重 分 散    */
float kajukyou;             /*    加 重 共 分 散  */
float a[3];                 /*   判別関数式の係数  */

char ff1[1000],ff2[1000],ff3[1000],ff4[1000],ff5[1000];
retu ichiretu;
FILE *fp,*fq,*fr,*fs,*ft,*fu;

char ransuu(char *,FILE *,FILE *,retu *);
float hkansuu(FILE *,FILE *,retu *,int *,float *);
float bunkyou(FILE *,FILE *,retu *,int *,float *,float *,float *);
float kajuuheikin(int *,float *,float *,float *,float *);
float hakidashi(float *,float *,float *,float *);
void hanbetubunseki(FILE *,FILE *,retu *,float *);



for(i=0;i<2;i++){
  sample[i]=0;
  heikin[i]=0.0;
  heikin[i+2]=0.0;
  bunsan[i]=0.0;
  bunsan[i+2]=0.0;
  kyoubunsan[i]=0.0;
  kajubun[i]=0.0;
}
kajukyou=0.0;
a[0]=0.0;
a[1]=0.0;
a[2]=0.0;

fputs("\n 正解データファイルの名前を入力して下さい。\n",stdout);
scanf("%s",ff1);
fputs("\n 不正解データファイルの名前を入力して下さい。\n",stdout);
scanf("%s",ff2);
fputs("\n",stdout);
strcpy(ff3,ff2);
strcat(ff3,"data");
```

```
  if((fp=fopen(ff1,"r"))==NULL){
    fputs("ファイル1 オープンエラー !!  なにしとんじゃい !! \n",stdout);
    exit(1);
  }
  else{
/*    if((fq=fopen(ff2,"r"))==NULL){
      fputs("ファイル2 オープンエラー !!  なにしとんじゃい !! \n",stdout);
      exit(1);
    }
    else{
*/
      if((fr=fopen(ff3,"w"))==NULL){
        fputs("ファイル3 オープンエラー !!  なにしとんじゃい !! \n",stdout);
        exit(1);
      }
      else{

        ransuu(ff2,fp,fr,&ichiretu);

fputs("何か入力して下さい\n",stdout);
getchar();

        freopen(ff1,"r",fp);
        freopen(ff3,"r",fr);

        hkansuu(fp,fr,&ichiretu,&sample,&heikin);

        printf("サンプル数 %d 平均1 %f 平均2 %f\n",sample[0],heikin[0],heikin[1]);
printf("サンプル数 %d 平均1 %f 平均2 %f\n",sample[1],heikin[2],heikin[3]);
printf("\n");

        freopen(ff1,"r",fp);
        freopen(ff3,"r",fr);

        bunkyou(fp,fr,&ichiretu,&sample,&heikin,&bunsan,&kyoubunsan);

        printf("分散1 %f 分散2 %f 共分散 %f\n",bunsan[0],bunsan[1],kyoubunsan[0]);
        printf("分散1 %f 分散2 %f 共分散 %f\n",bunsan[2],bunsan[3],kyoubunsan[1]);
printf("\n");

        kajuuheikin(&sample,&bunsan,&kyoubunsan,&kajubun,&kajukyou);

        printf("プール後の分散1 %f 分散2 %f 共分散 %f\n",kajubun[0],kajubun[1],kajukyou);
```

```
        hakidashi(&heikin,&kajubun,&kajukyou,&a);

        printf("求められた係数 a[0] = %f a[1] = %f a[2] = %f\n\n",a[0],a[1],a[2]);

        freopen(ff1,"r",fp);
strcpy(ff5,ff1);
strcat(ff5,".");
strcat(ff5,ff2);
strcat(ff5,".kekka");
        if((fu=fopen(ff5,"w"))==NULL){
          fputs("ファイル オープンエラー !!  なにしとんじゃい !! \n",stdout);
          exit(1);
}
        else{
          hanbetubunseki(fp,fu,&ichiretu,&a);
}fclose(fu);

        if((fs=fopen(ff2,"r"))==NULL){
          fputs("ファイル オープンエラー !!  なにしとんじゃい !! \n",stdout);
          exit(1);
        }
        else{
          strcpy(ff4,ff2);
  strcat(ff4,".");
  strcat(ff4,ff1);
          strcat(ff4,".kekka");
          if((ft=fopen(ff4,"w"))==NULL){
            fputs("ファイル オープンエラー !!  なにしとんじゃい !! \n",stdout);
            exit(1);
  }
          else{
            hanbetubunseki(fs,ft,&ichiretu,&a);
  }fclose(ft);
       }fclose(fs);
     }fclose(fr);
  }fclose(fp);
}



/*-------------------------------------------------------------
```

元のデータを idiom と general に分ける関数

```
  ------------------------------------------------------------------*/
/*
void senbetu(                    ){
  char idiom[1000];
  FILE *fp,*fq,*fr,*fs;
  if((fp=fopen("         ","r"))==NULL){
    fputs("File Open Error !!  なにしとんじゃい !! \n",stdout);
    exit(1);
  }
  else{
    if((fq=fopen("         ","r"))==NULL){
      fputs("File Open Error !!  なにしとんじゃい !! \n",stdout);
      exit(1);
    }
    else{
      fr=fopen("       ","w");
      fs=fopen("       ","w");
      if((fscanf(fq,"%s",idiom))!=EOF && fscanf(fp,"%s",


      fclose(fs);
      fclose(fr);
    }fclose(fq);
  }fclose(fp);
}
*/
/*
typedef struct{
  char tango[1000];
  char mae[1000];
  char ato[1000];
  int kazu;
  int kazum;
  int kazua;
  float kazumh;
  float kazuah;
  float kazuw;
} retu;
*/
```

```
/*-------------------------------------------------------------

                    乱　数　発　生　用　関　数

  -------------------------------------------------------------*/

char ransuu(char *ff2,FILE *fp,FILE *fr,retu *ichiretu){
  int gyou;       /*   gyou は、学習データ不正解の総数     */
  int *data[100000],i,j,k,ncha;
  char tango[10000],mojiretu[10000];

  FILE *fq;

  float retuget(FILE *,retu *);


  i=0;
  j=0;
  k=0;
  gyou=0;
  ncha=0;

  while((fgets(tango,10000,fp))!=NULL){
    i++;          /* ========= 学習データ正解版の数 =========== */
  }

  i*=6;           /* ========= 学習データ不正解版の数 ========== */

  if((fq=fopen(ff2,"r"))==NULL){
    fputs("File Open Error !! なにしとんじゃい !! (ransuu 1)\n",stdout);
    exit(1);
  }
  else{
    while((fscanf(fq,"%s",(*ichiretu).tango))!=EOF){
      retuget(fq,ichiretu);
      gyou+=1;    /* ======== 学習データ不正解の総数の検出部 ======= */
    }

        /*=====     乱　数　ジ　ェ　ネ　レ　ー　タ　ー      =====*/
    while(j<i){
      data[j]=(int *)malloc(sizeof(int));
      *data[j]=1+(int)(gyou*(float)(rand())/(RAND_MAX+1.0));
      j+=1;
    }
```

```
        /*======   こ こ か ら 乱 数 の ソ ー ト   =======*/
    for(j=0;j<i-1;j++){
      for(k=j+1;k<i;k++){
        if(*data[j]>*data[k]){
          ncha=*data[j];
  *data[j]=*data[k];
  *data[k]=ncha;
        }                  /*  同じ数は除く  */
        else if(*data[j]==*data[k]){
  *data[k]=*data[i-1];
  i-=1;
        }
      }
    }
    gyou=0;
    j=0;
    if((freopen(ff2,"r",fq))==NULL){
      fputs("File Open Error !!  なにしとんじゃい !! (ransuu 2)\n",stdout);
      exit(1);
    }
    else{
      while(j<i){
        fgets(mojiretu,10000,fq);
        gyou+=1;
        if(gyou==*data[j]){
  fputs(mojiretu,fr);
          j++;
        }
      }
    }
  }fclose(fq);
}



/*-------------------------------------------------------------

            平  均  を  求  め  る  関  数

  -------------------------------------------------------------*/

float hkansuu(FILE *fp,FILE *fq,retu *ichiretu,int *sample,float *heikin){
```

```
    float retuget(FILE *,retu *);
                                        /* サンプル1 総数及び平均 */

    while((fscanf(fp,"%s",(*ichiretu).tango)!=EOF)){
      retuget(fp,ichiretu);
      sample[0]+=1;
      heikin[0]+=(*ichiretu).kazumh;
      heikin[1]+=(*ichiretu).kazuah;
    }
    heikin[0]/=sample[0];
    heikin[1]/=sample[0];
                                        /* サンプル2 総数及び平均 */

    while((fscanf(fq,"%s",(*ichiretu).tango)!=EOF)){
      retuget(fq,ichiretu);
      sample[1]+=1;
      heikin[2]+=(*ichiretu).kazumh;
      heikin[3]+=(*ichiretu).kazuah;
    }
    heikin[2]/=sample[1];
    heikin[3]/=sample[1];
}



/*-----------------------------------------------------------------

    スキャン部 最初の変数はファイル管理のため他で管理している。

    -----------------------------------------------------------------*/

float retuget(FILE *ff,retu *ichiretu){
    fscanf(ff,"%s",(*ichiretu).mae);
    fscanf(ff,"%s",(*ichiretu).ato);
    fscanf(ff,"%d",&(*ichiretu).kazu);
    fscanf(ff,"%d",&(*ichiretu).kazum);
    fscanf(ff,"%d",&(*ichiretu).kazua);
    fscanf(ff,"%f",&(*ichiretu).kazumh);
    fscanf(ff,"%f",&(*ichiretu).kazuah);
    fscanf(ff,"%f",&(*ichiretu).kazuw);
}
```

```
/*-----------------------------------------------------------------

        分　散　・　共　分　散　を　求　め　る　関　数

  -----------------------------------------------------------------*/

float bunkyou(FILE *fp,FILE *fq,retu *ichiretu,int *sample,float *heikin,float *bunsan,f
  float retuget(FILE *,retu *);
                                    /*サンプル1 分散・共分散 */

  while((fscanf(fp,"%s",(*ichiretu).tango)!=EOF)){
    retuget(fp,ichiretu);
    bunsan[0]+=((*ichiretu).kazumh-heikin[0])*((*ichiretu).kazumh-heikin[0]);
    bunsan[1]+=((*ichiretu).kazuah-heikin[1])*((*ichiretu).kazuah-heikin[1]);
    kyoubunsan[0]+=((*ichiretu).kazumh-heikin[0])*((*ichiretu).kazuah-heikin[1]);
  }
  bunsan[0]/=sample[0];
  bunsan[1]/=sample[0];
  kyoubunsan[0]/=sample[0];
                                    /*サンプル2 分散・共分散 */

  while((fscanf(fq,"%s",(*ichiretu).tango)!=EOF)){
    retuget(fq,ichiretu);
    bunsan[2]+=((*ichiretu).kazumh-heikin[2])*((*ichiretu).kazumh-heikin[2]);
    bunsan[3]+=((*ichiretu).kazuah-heikin[3])*((*ichiretu).kazuah-heikin[3]);
    kyoubunsan[1]+=((*ichiretu).kazumh-heikin[2])*((*ichiretu).kazuah-heikin[3]);
  }
  bunsan[2]/=sample[1];
  bunsan[3]/=sample[1];
  kyoubunsan[1]/=sample[1];
}



/*-----------------------------------------------------------------

        分　散　・　共　分　散　の　加　重　平　均

  -----------------------------------------------------------------*/

float kajuuheikin(int *sample,float *bunsan,float *kyoubunsan,float *kajubun,float *kaju
```

```
  kajubun[0]=(sample[0]*bunsan[0]+sample[1]*bunsan[2])/(sample[0]+sample[1]);
  kajubun[1]=(sample[0]*bunsan[1]+sample[1]*bunsan[3])/(sample[0]+sample[1]);
  *kajukyou=(sample[0]*kyoubunsan[0]+sample[1]*kyoubunsan[1])/(sample[0]+sample[1]);
}



/*-------------------------------------------------------------

                    掃　き　出　し　法

  -------------------------------------------------------------*/

float hakidashi(float *heikin,float *kajubun,float *kajukyou,float *a){
  a[2]=((kajubun[0]*(heikin[1]-heikin[3]))-((*kajukyou)*(heikin[0]-heikin[2])))/(kajubun
  a[1]=((heikin[0]-heikin[2])-a[2]*(*kajukyou))/kajubun[0];
  a[0]=(a[1]*(heikin[0]+heikin[2])+a[2]*(heikin[1]+heikin[3]))/(-2);
}



/*-------------------------------------------------------------

                    判　別　処　理

  -------------------------------------------------------------*/

void hanbetubunseki(FILE *fr,FILE *fs,retu *ichiretu,float *a){
  float z;
  float retuget(FILE *,retu *);
  while((fscanf(fr,"%s",(*ichiretu).tango)!=EOF)){
    retuget(fr,ichiretu);
                              /* 線形判別方程式 */

    z=a[0]+a[1]*((*ichiretu).kazumh)+a[2]*((*ichiretu).kazuah);

    if(z>0){
      fprintf(fs,"%s\n",(*ichiretu).tango);
    }
  }
}
```

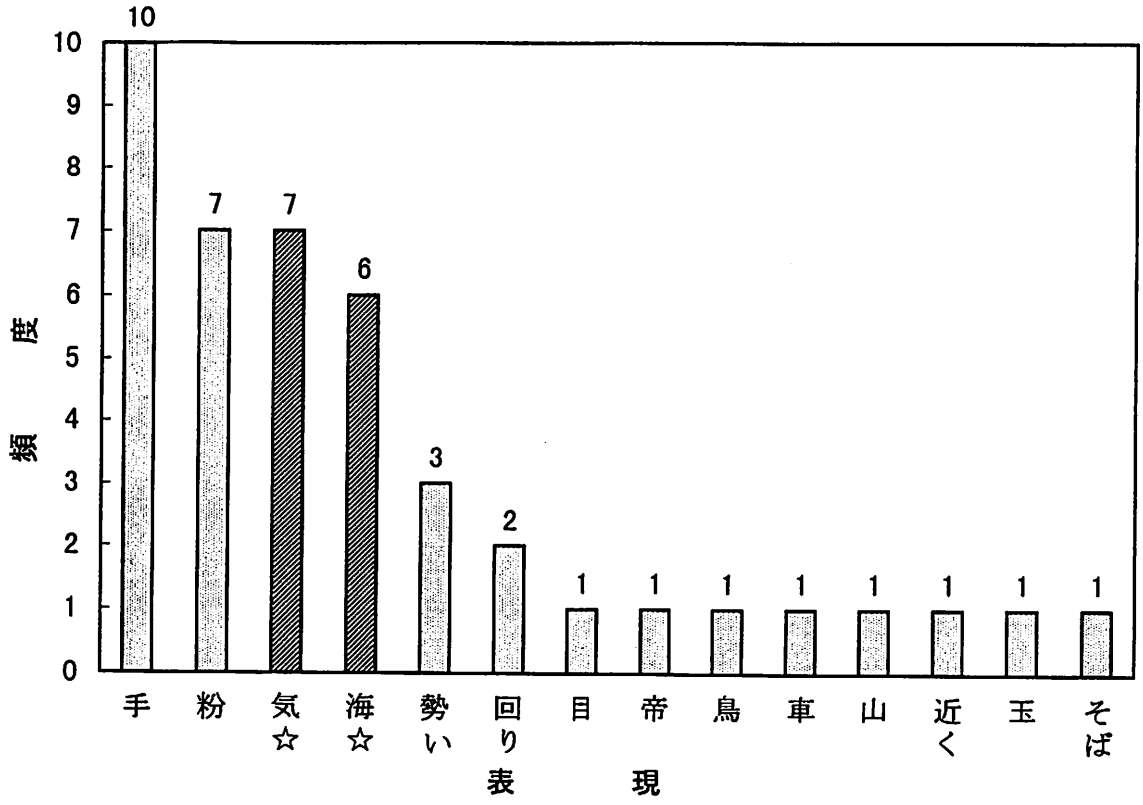
Appendix B

未抽出の慣用表現の分析グラフ

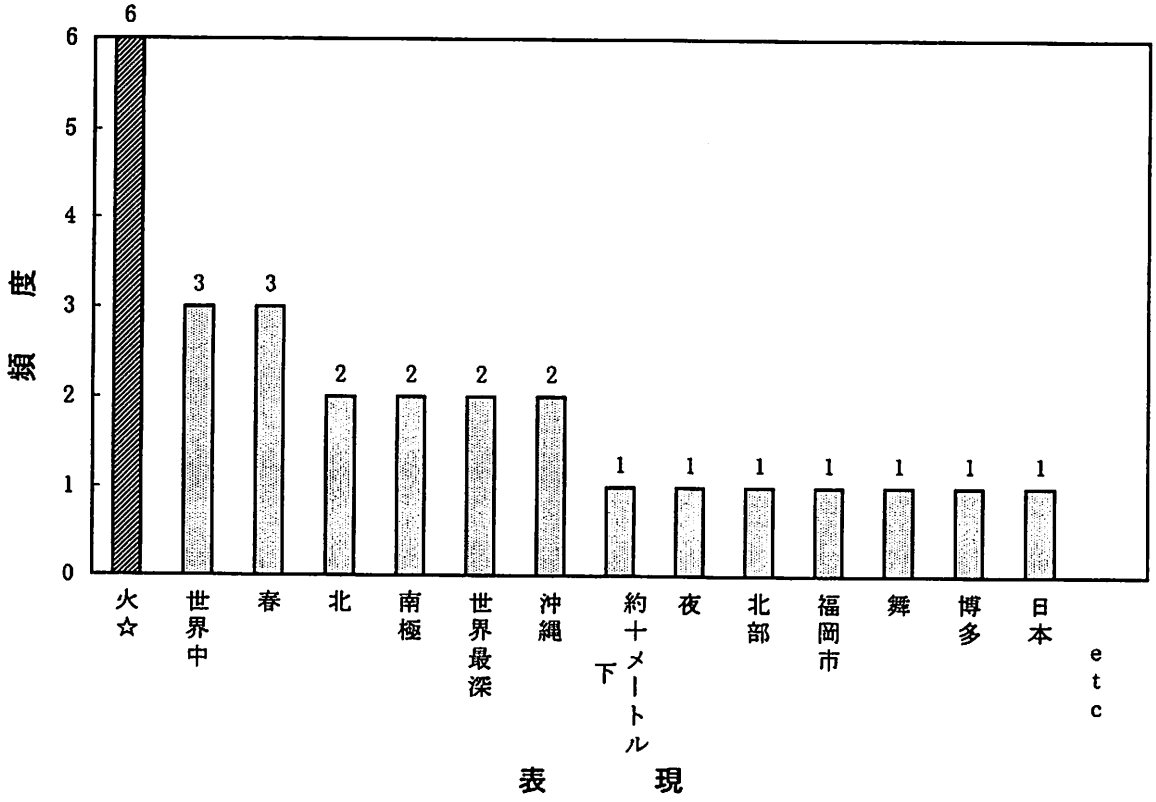
未抽出(1)

〔火の海〕

〔火　の　～〕



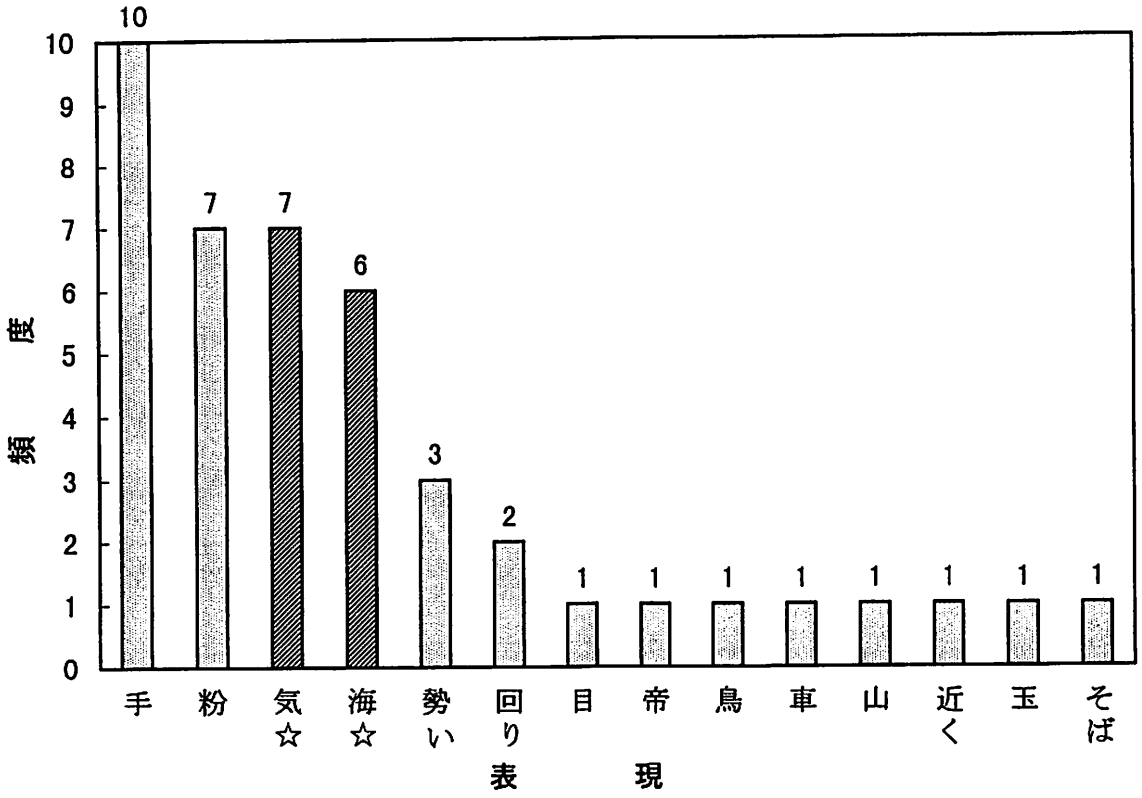
〔～　の　海〕



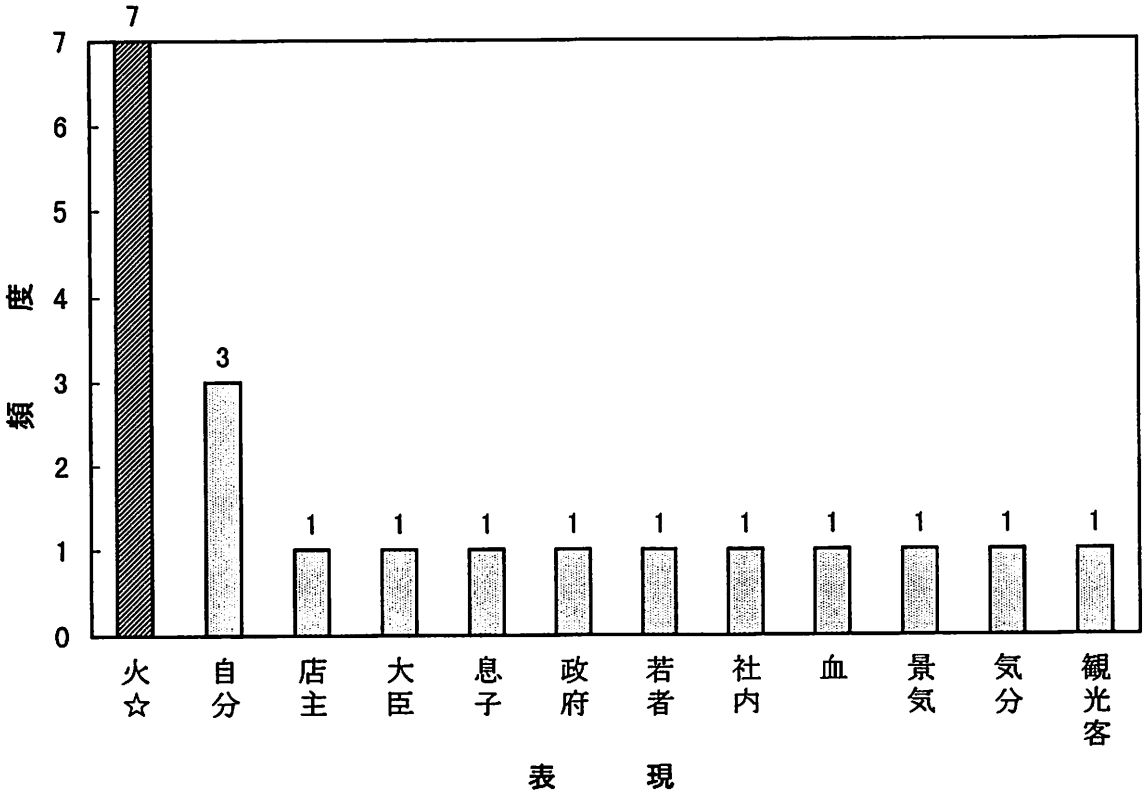
未抽出(2)

〔火の気〕

〔火　の　～〕



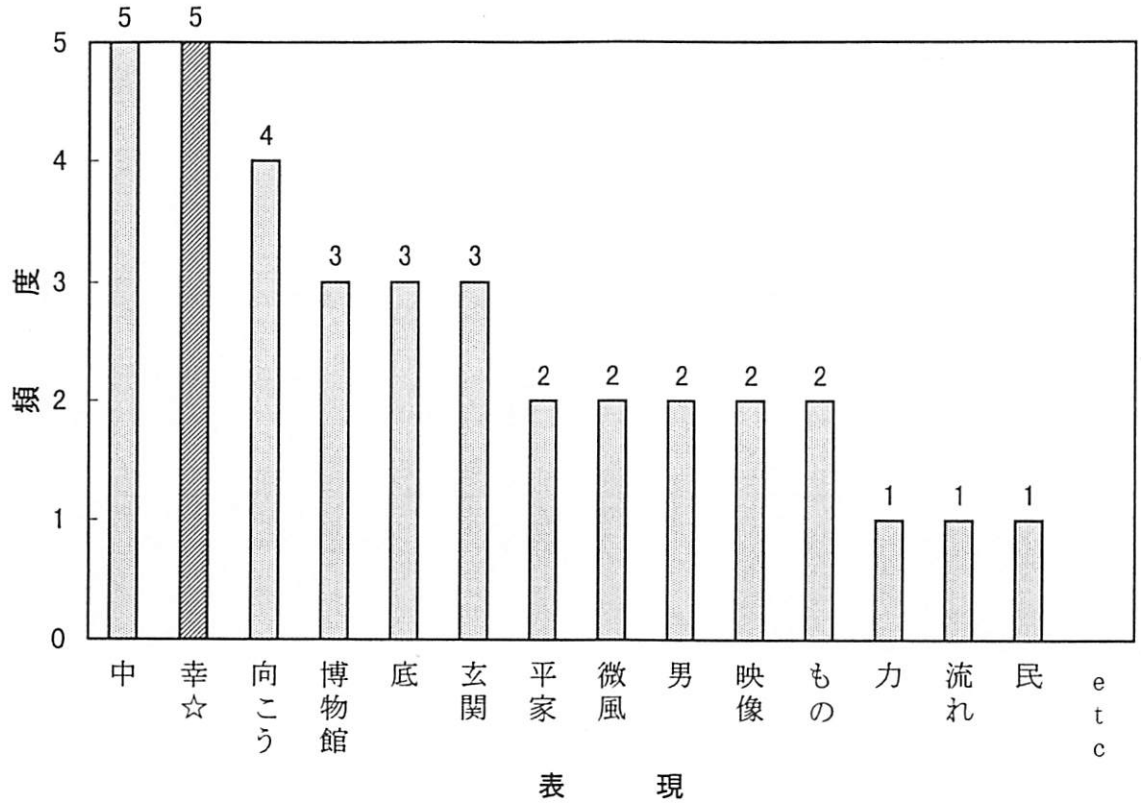
〔～　の　気〕



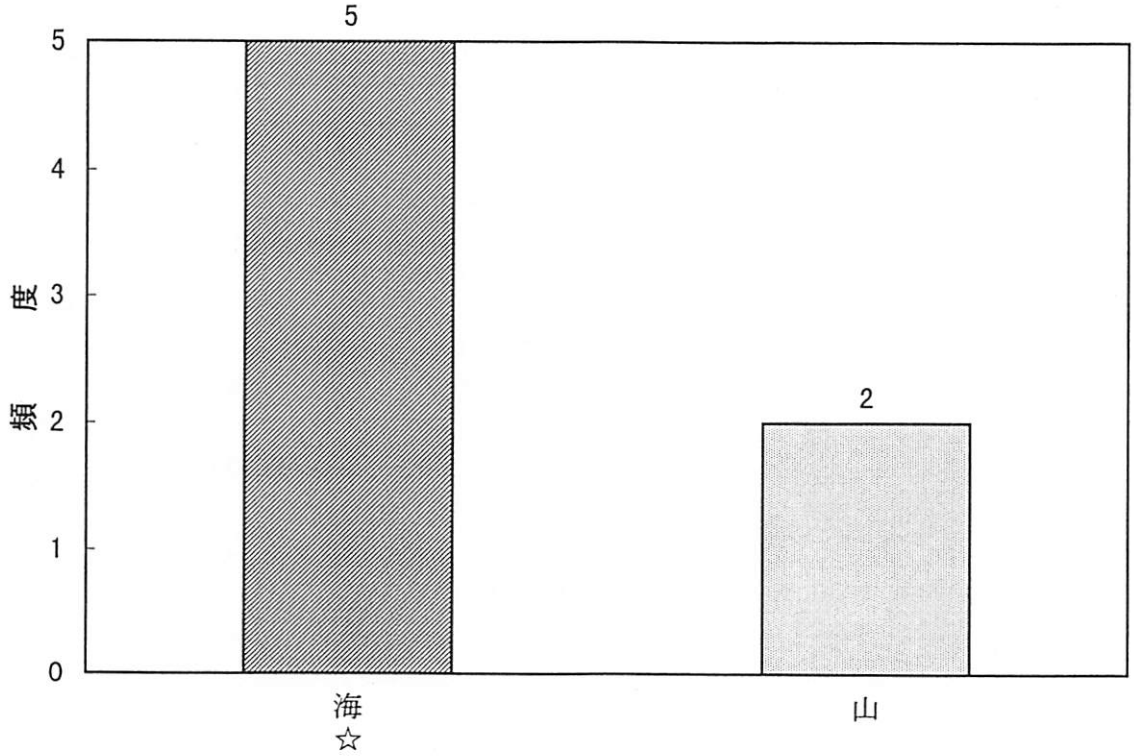
未抽出(3)

〔海の幸〕

〔海　の　～〕



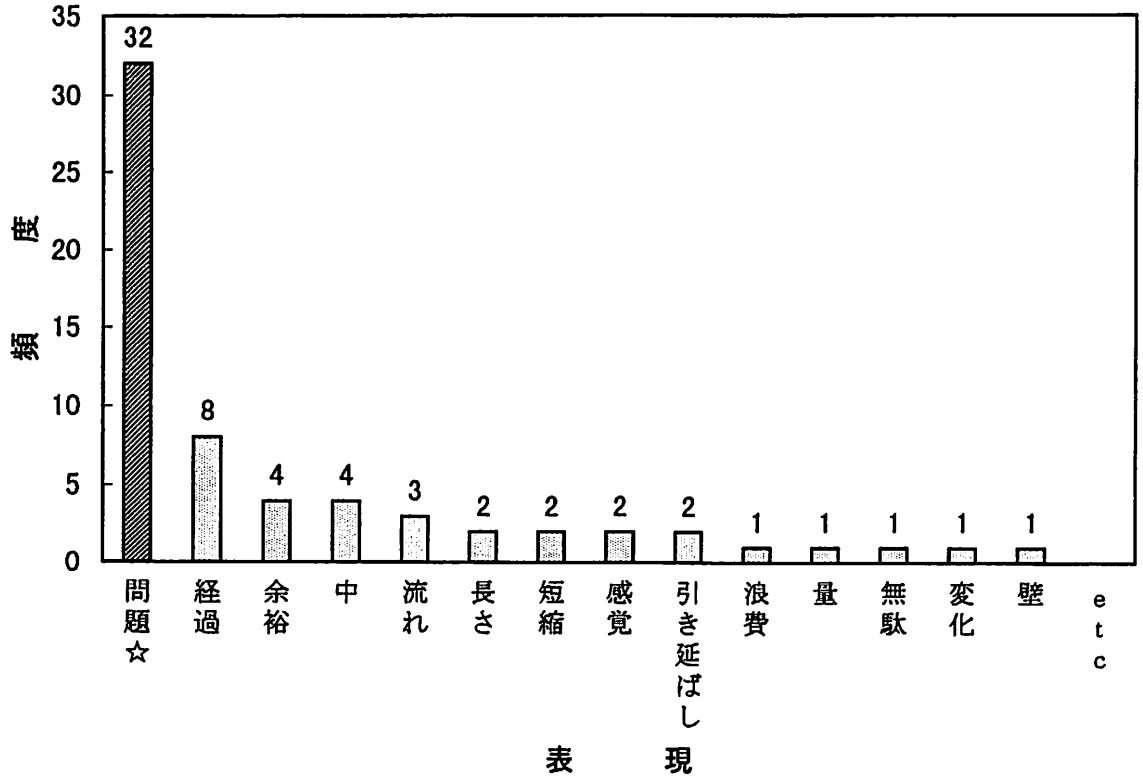
〔～　の　幸〕



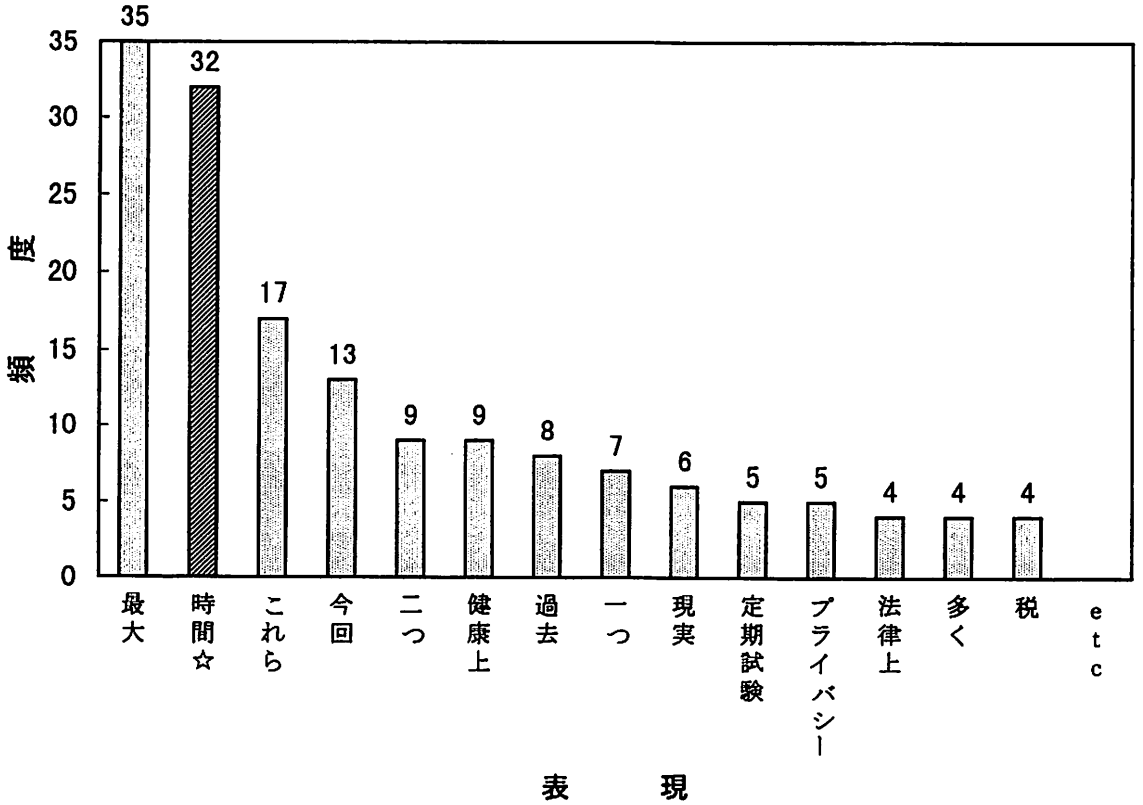
未抽出(4)

〔時間の問題〕

〔時　間　の　～〕



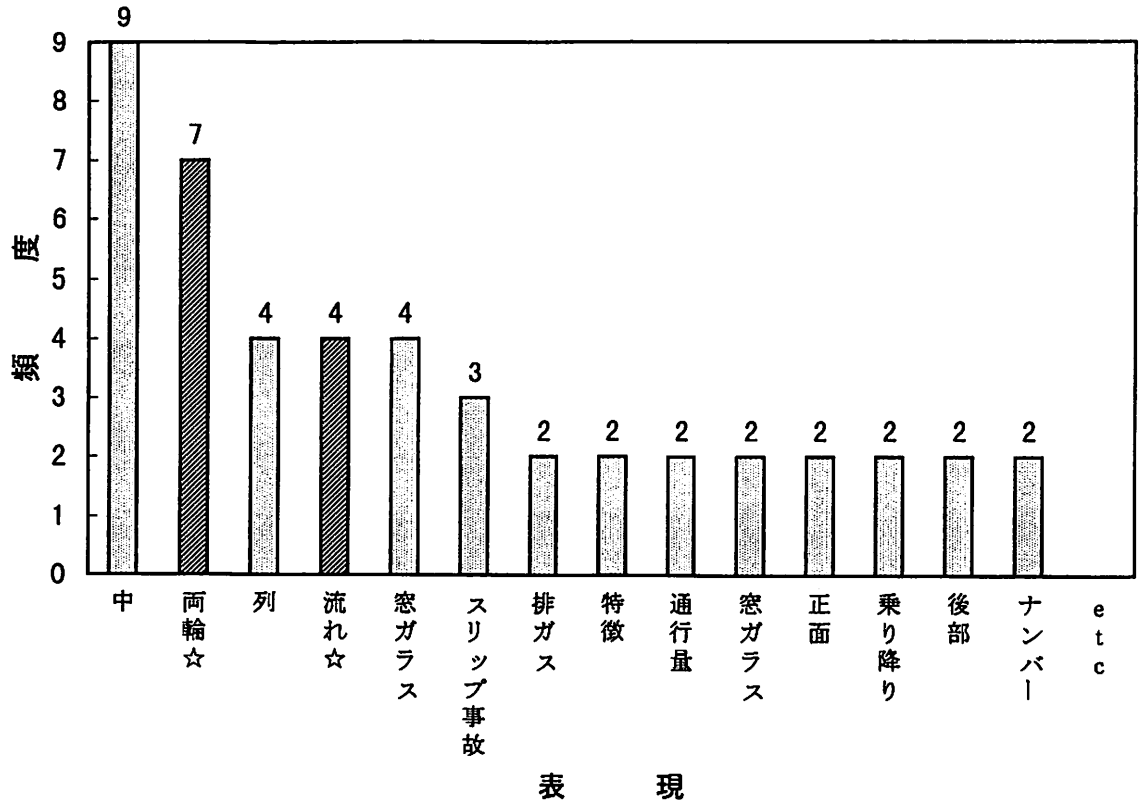
〔～　の　問　題〕



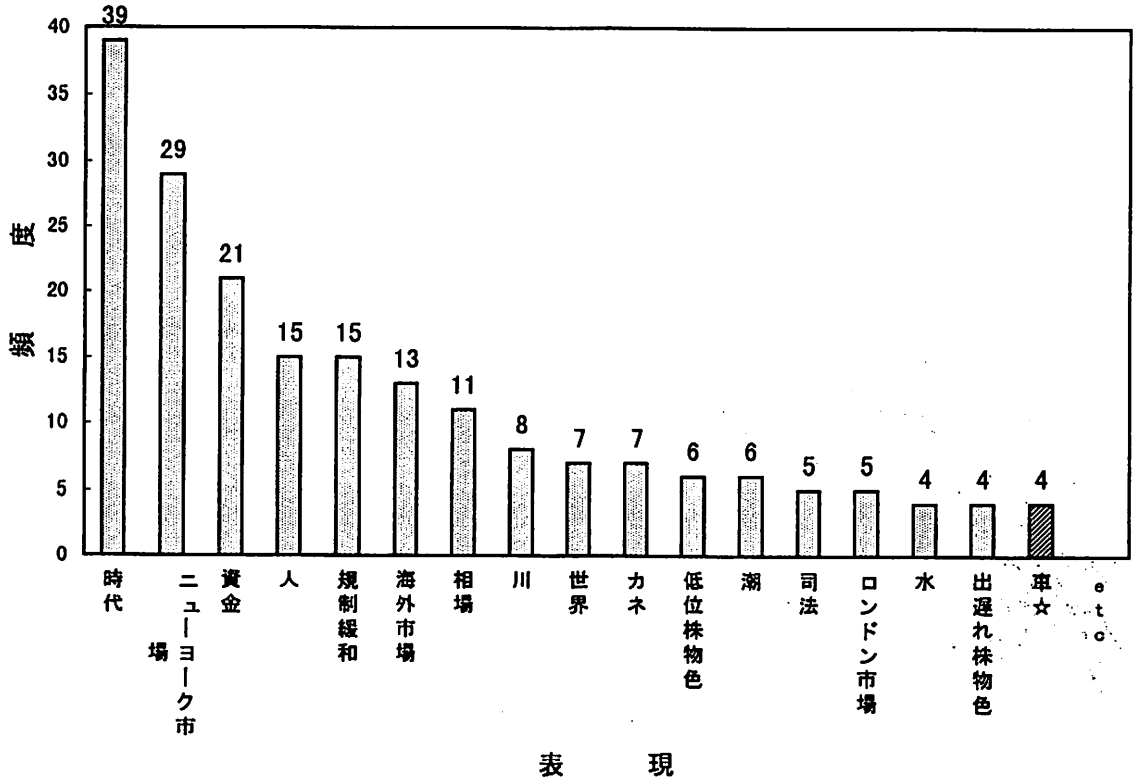
未抽出(5)

〔車の流れ〕

〔車　の　～〕



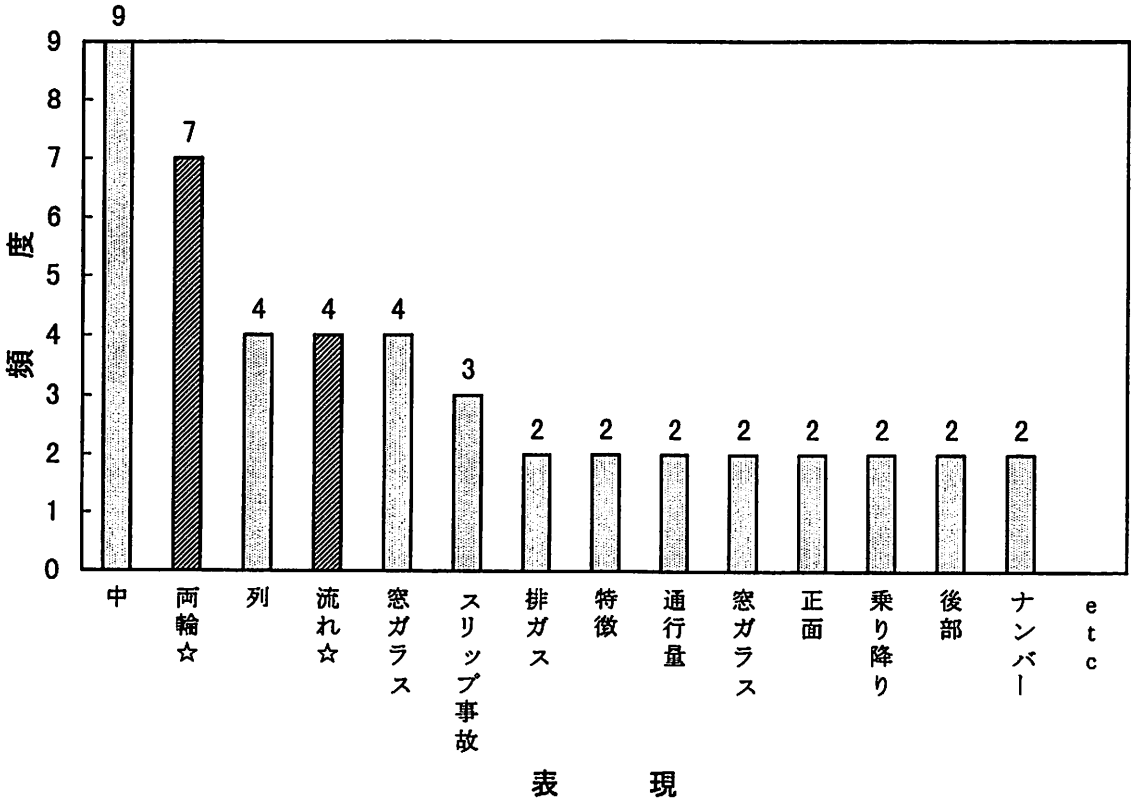
〔～　の　流　れ〕



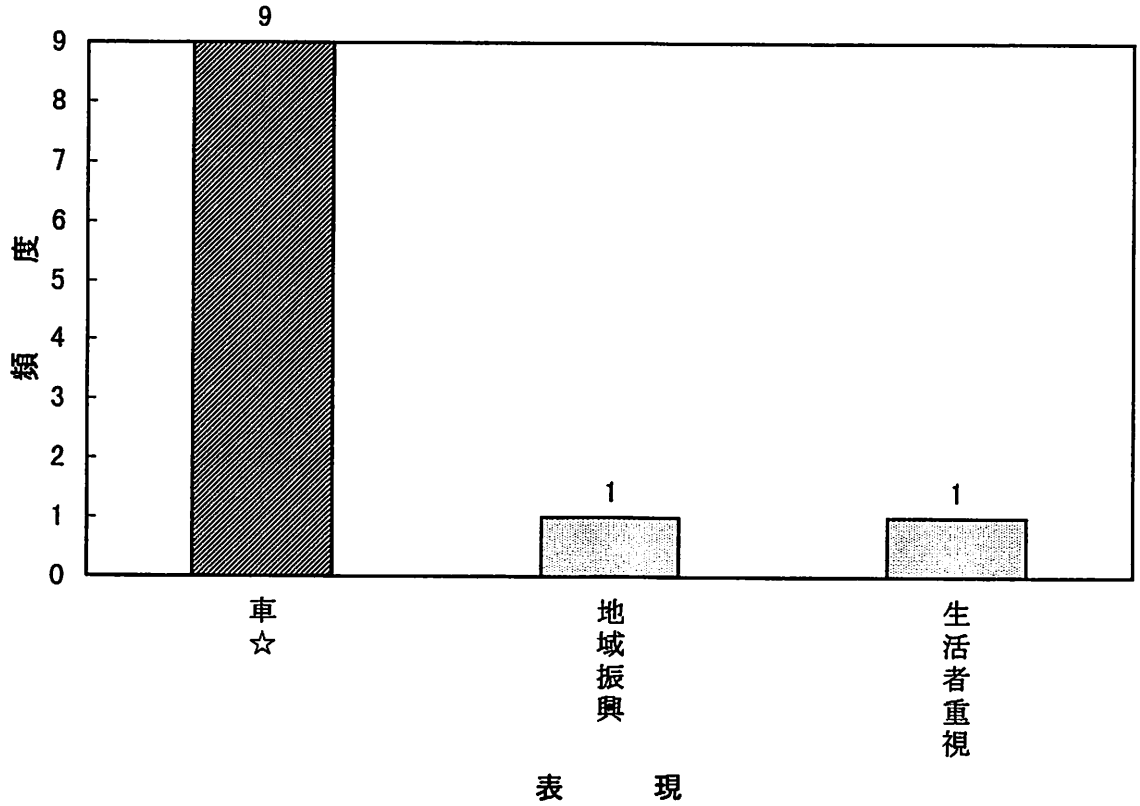
未抽出(6)

〔車の両輪〕

〔車　の　～〕



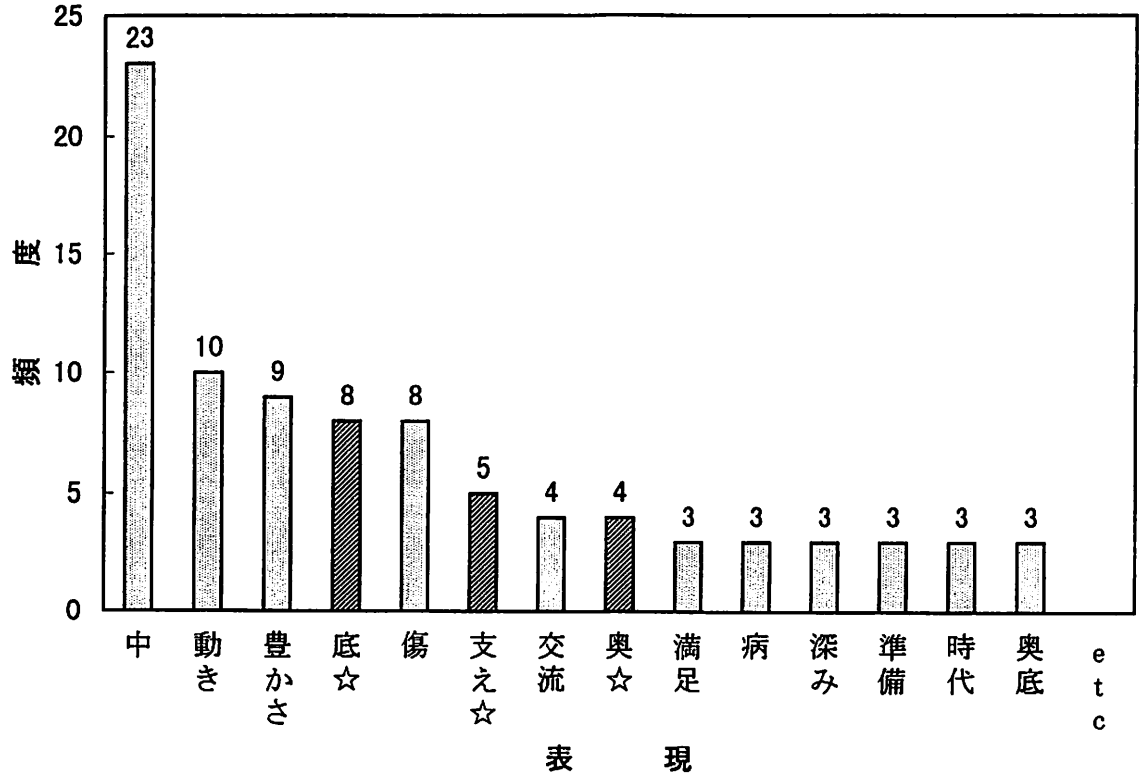
〔～　の　両　輪〕



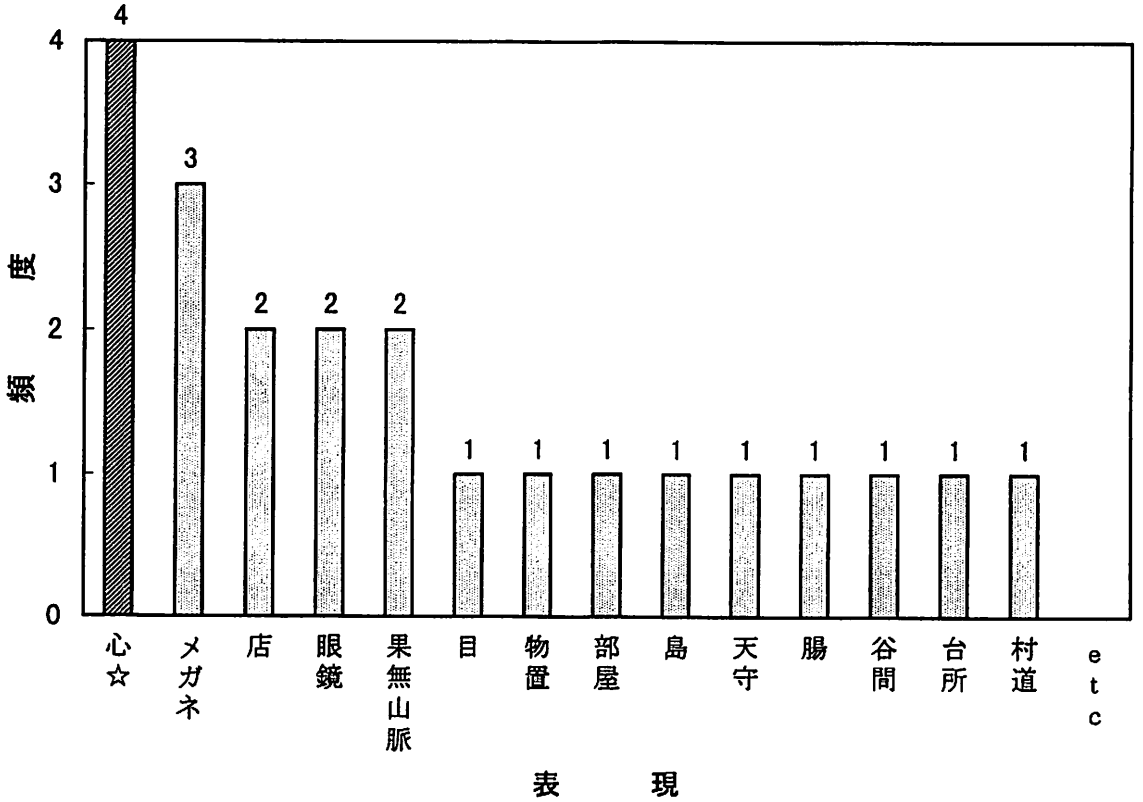
未抽出(7)

〔心の奥〕

〔心　の　～〕



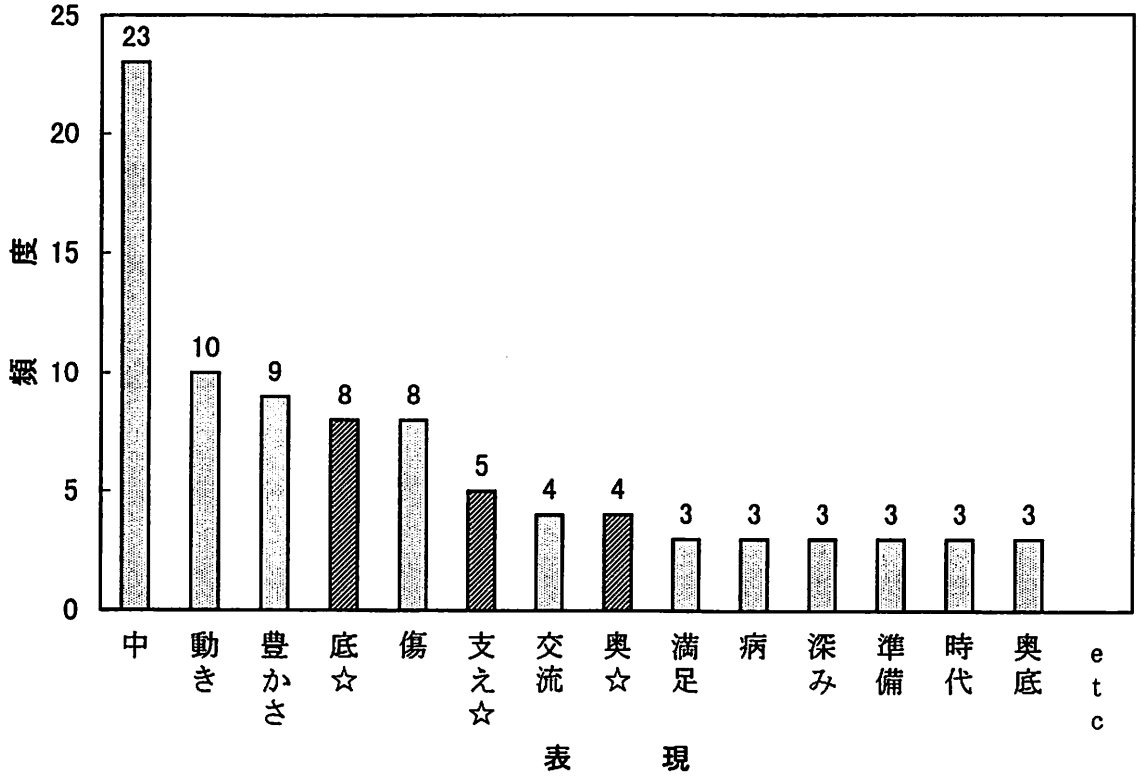
〔～　の　奥〕



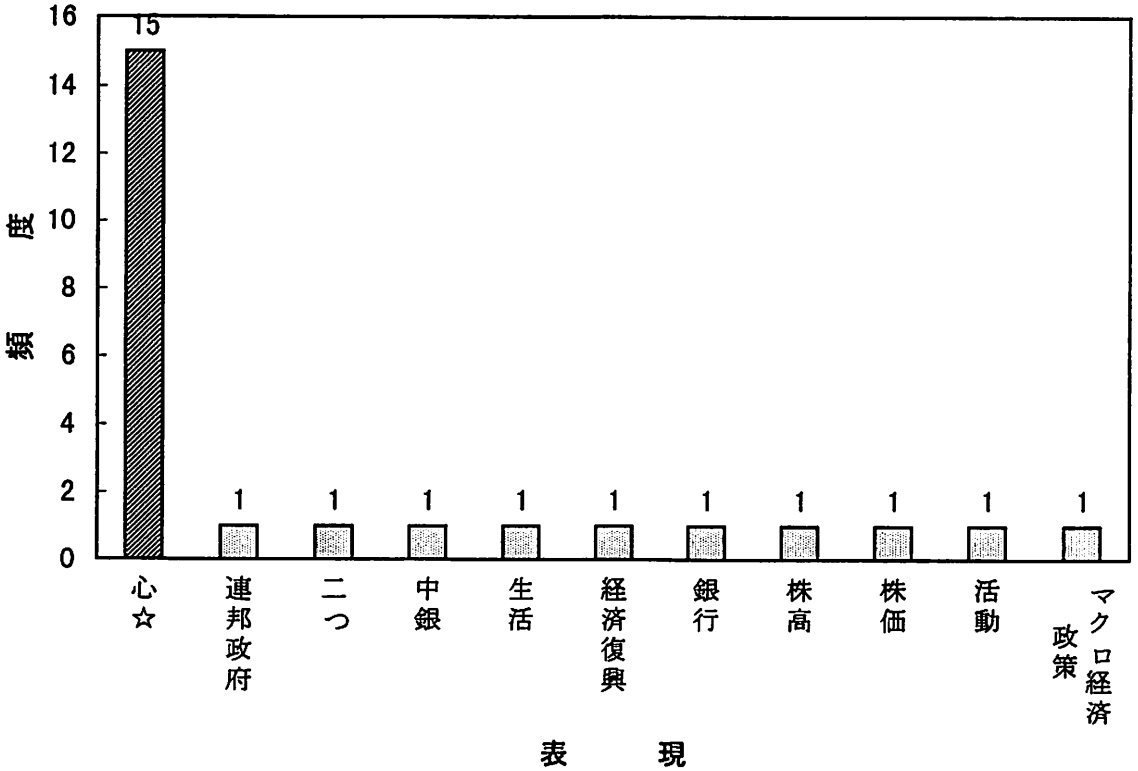
未抽出(8)

〔心の支え〕

〔心　の　～〕



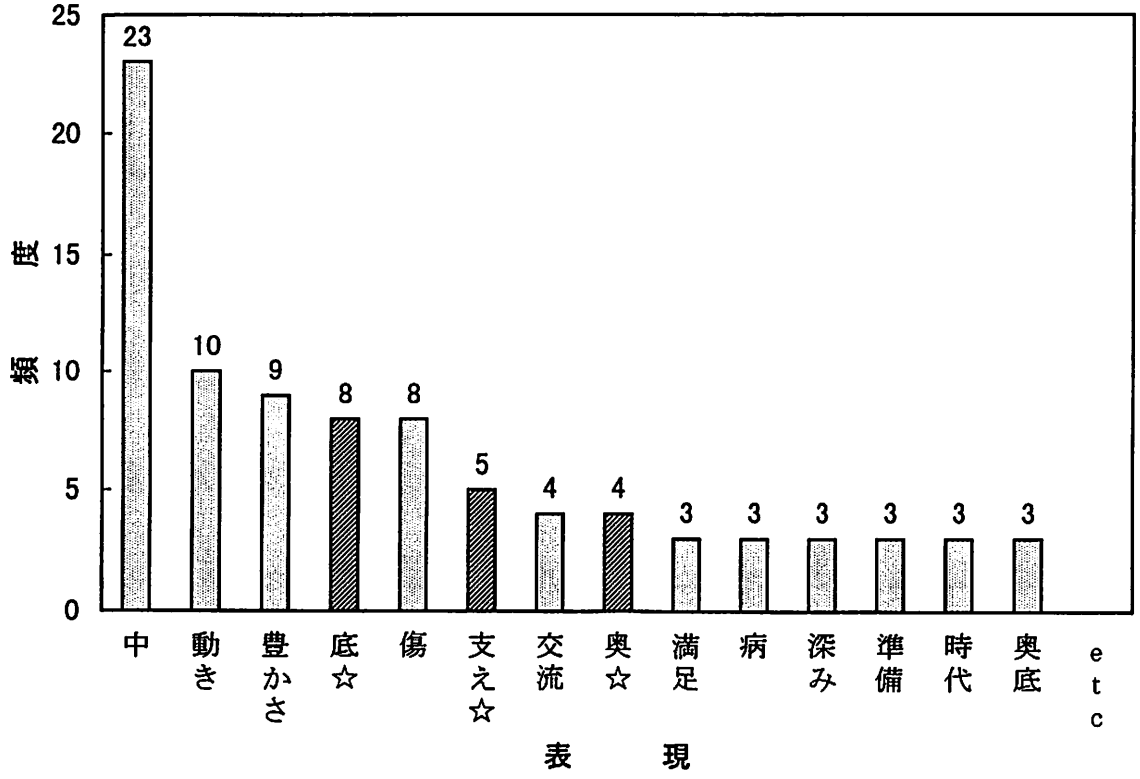
〔～　の　支　え〕



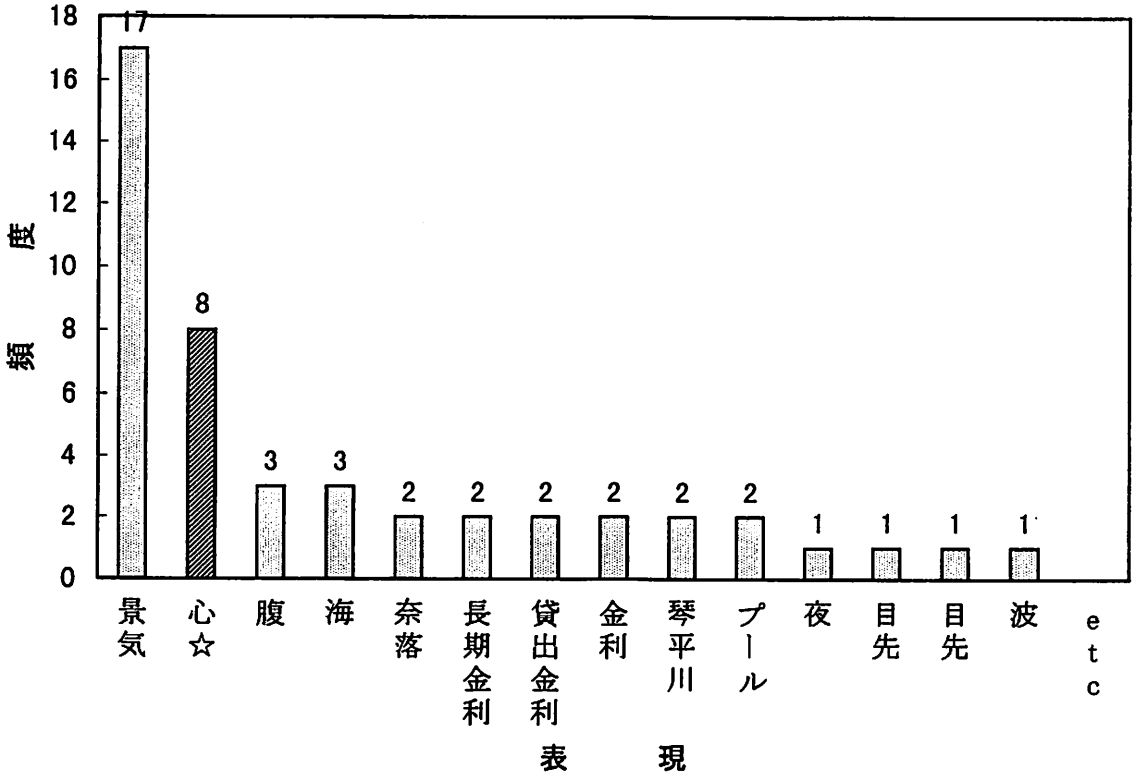
未抽出(9)

「心の底」

〔心　の　～〕



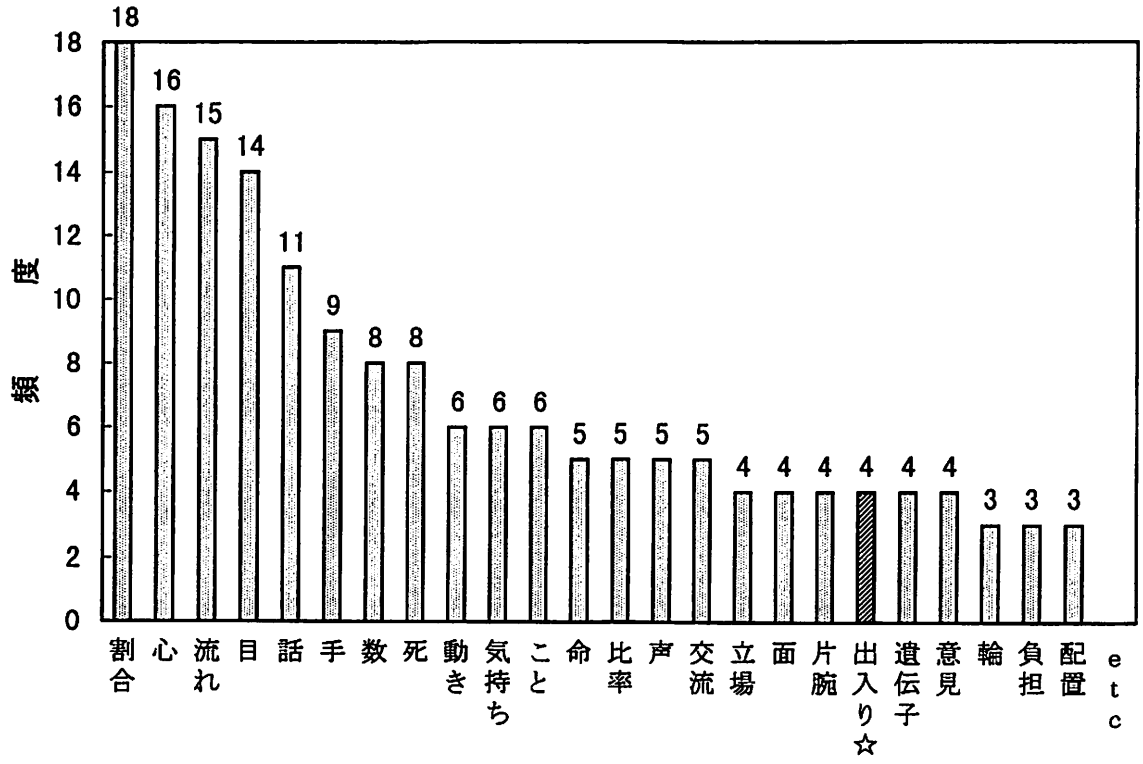
〔～　の　底〕



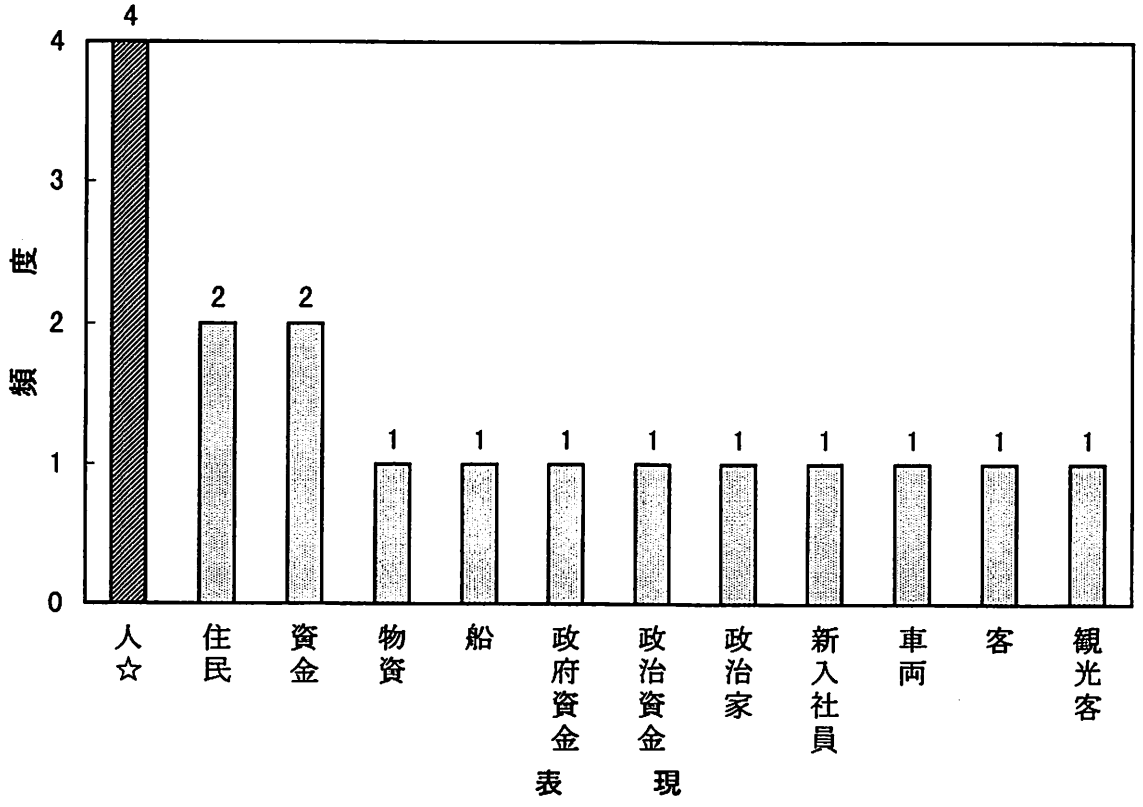
未抽出(10)

〔人の出入り〕

〔人　の　～〕



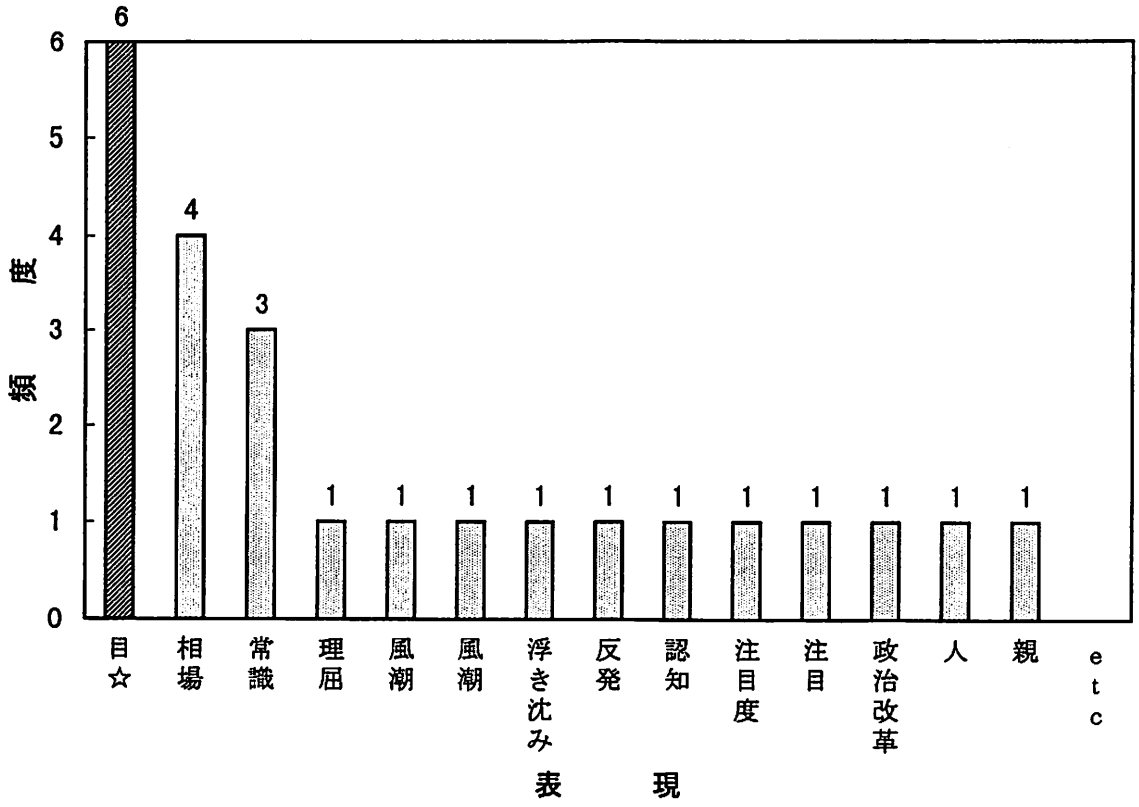
〔～　の　出　入　り〕



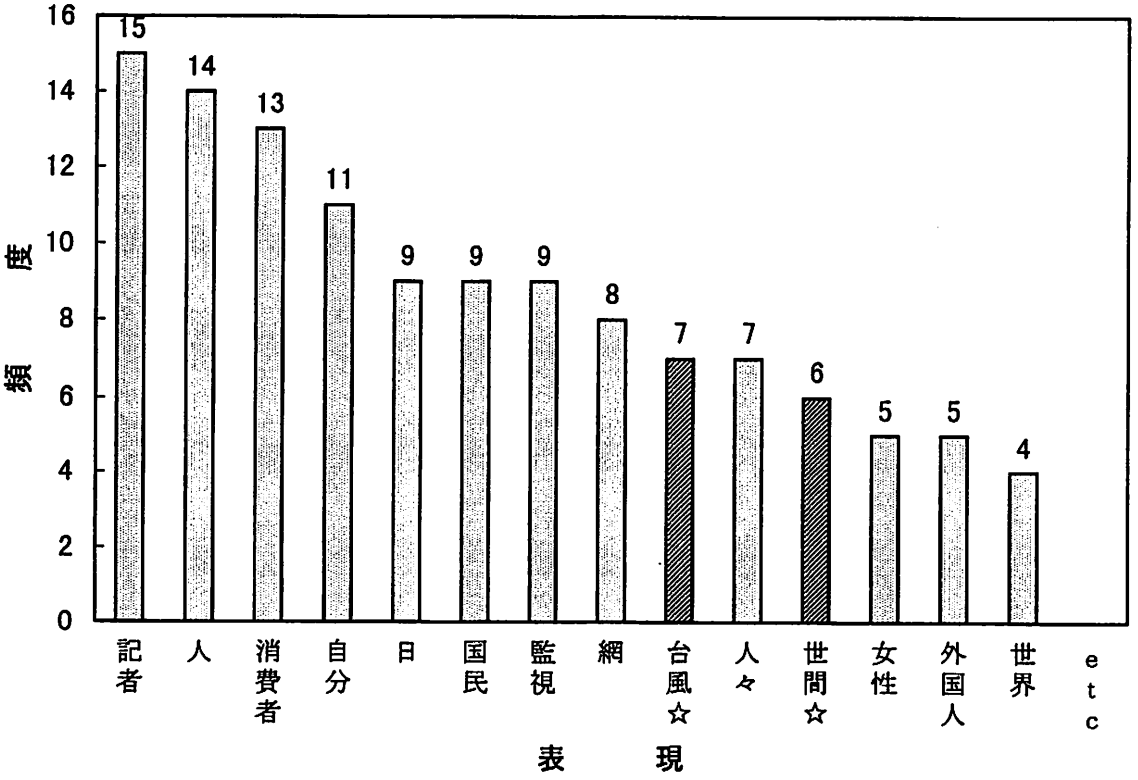
未抽出(11)

〔世間の目〕

〔世　間　の　～〕



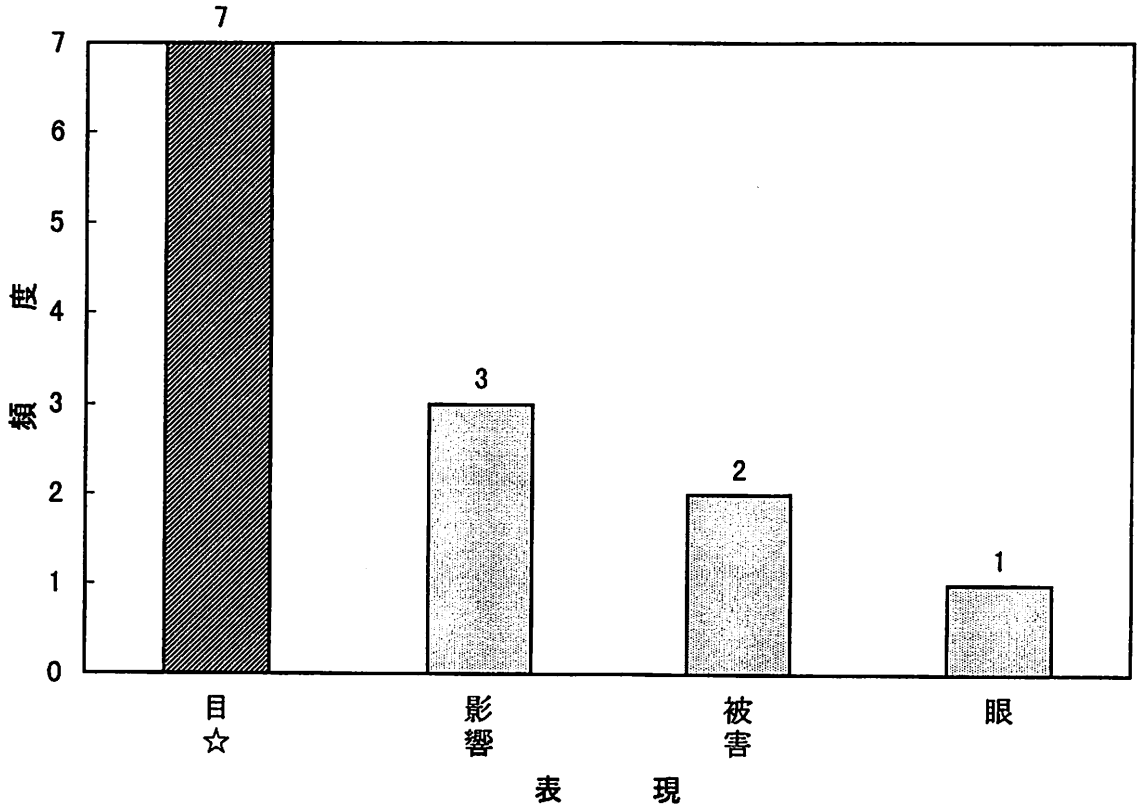
〔～　の　目〕



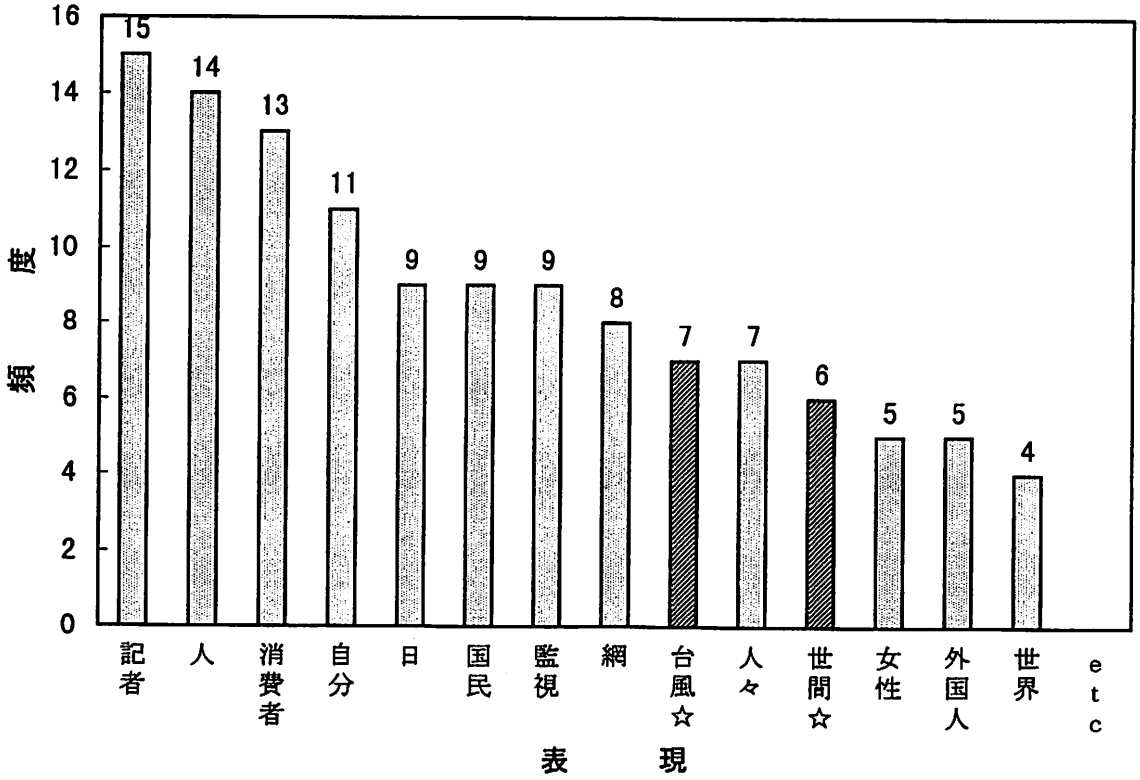
未抽出(12)

「台風の目」

〔台　　風　　の　　～〕



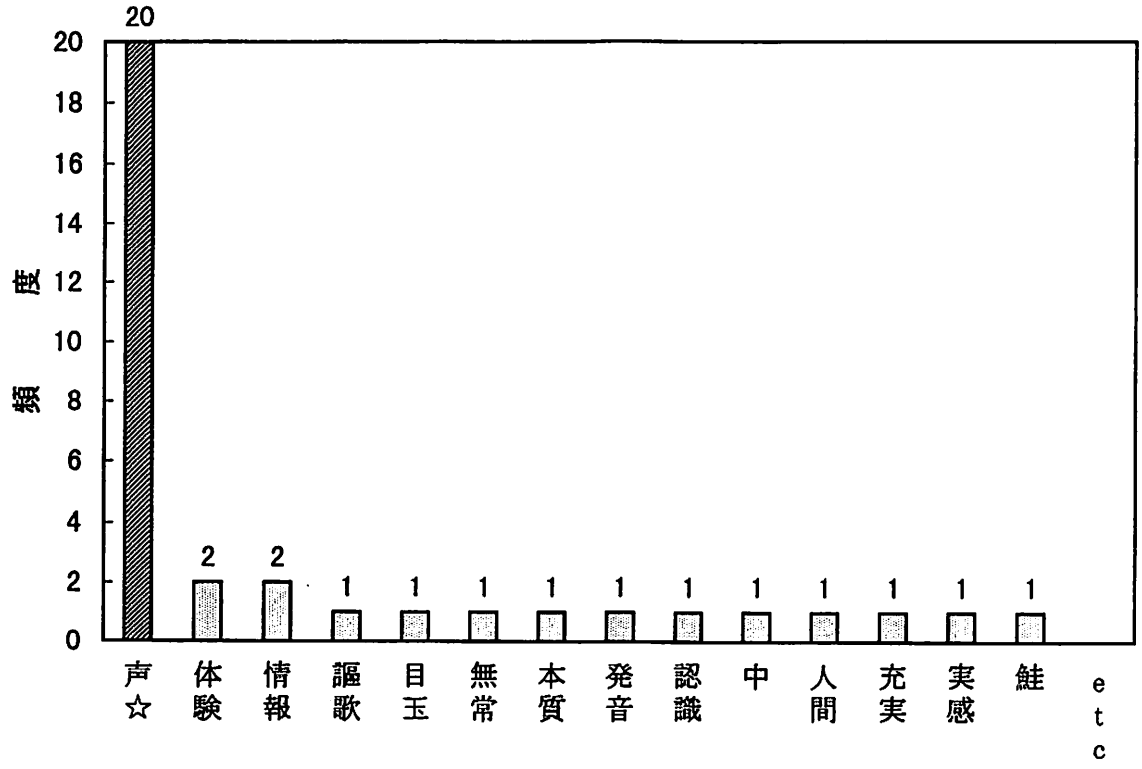
〔～　　の　　目〕



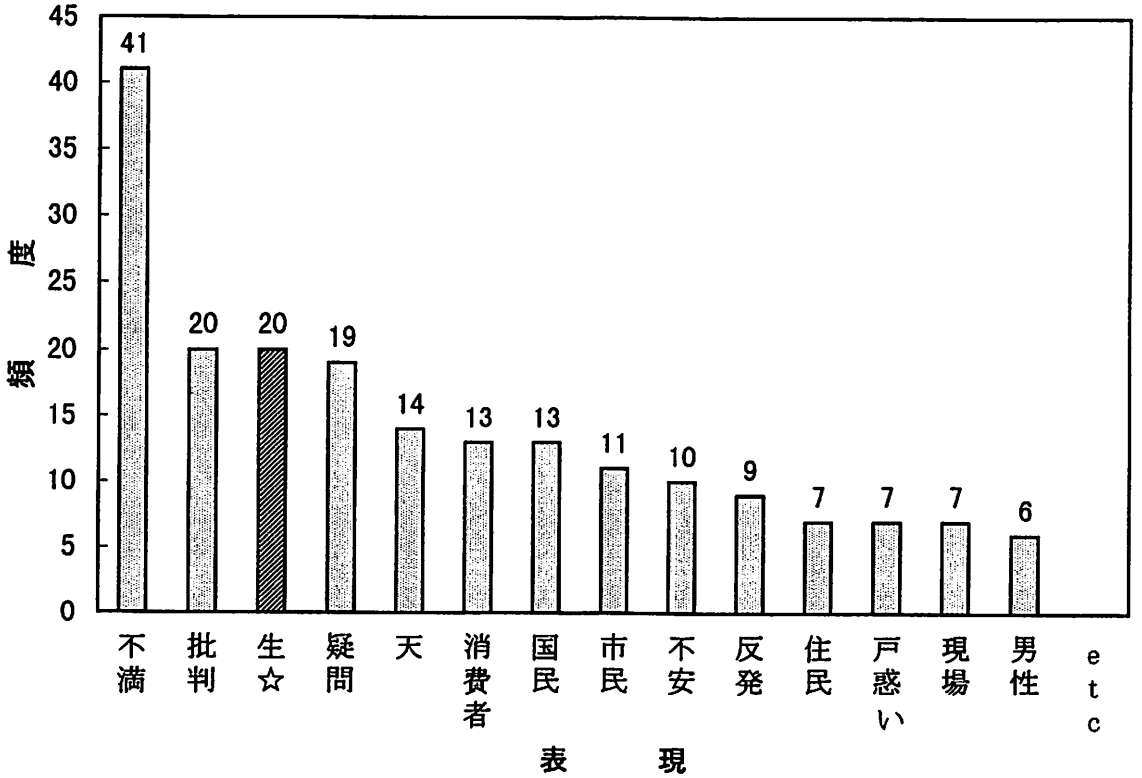
未抽出(13)

「牛の声」

〔生　の　～〕



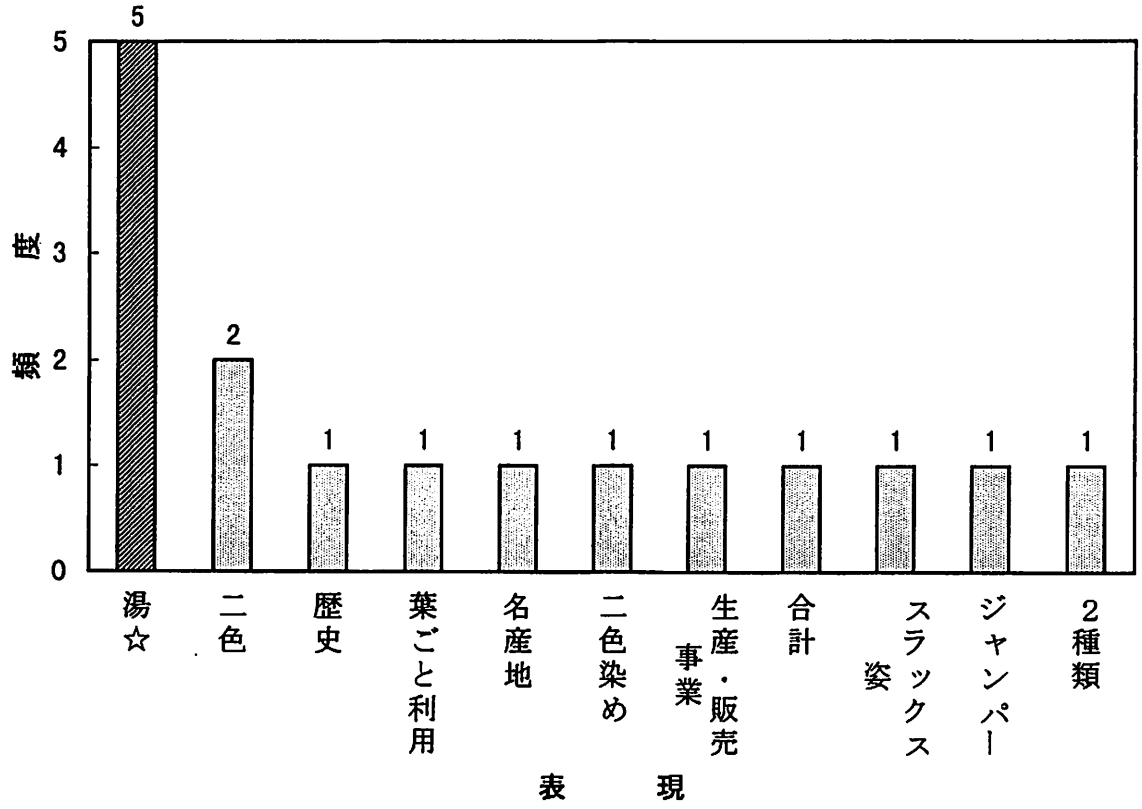
〔～　の　声〕



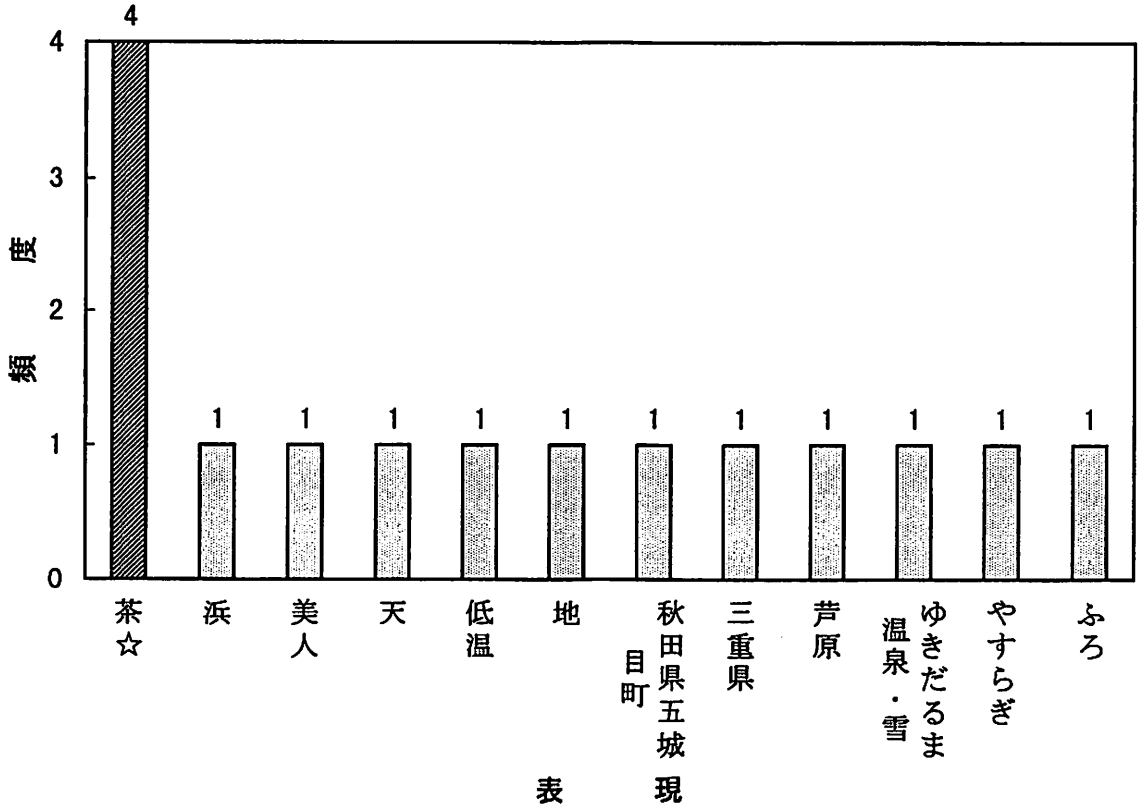
未抽出(14)

〔茶の湯〕

〔茶　の　～〕



〔～　の　湯〕



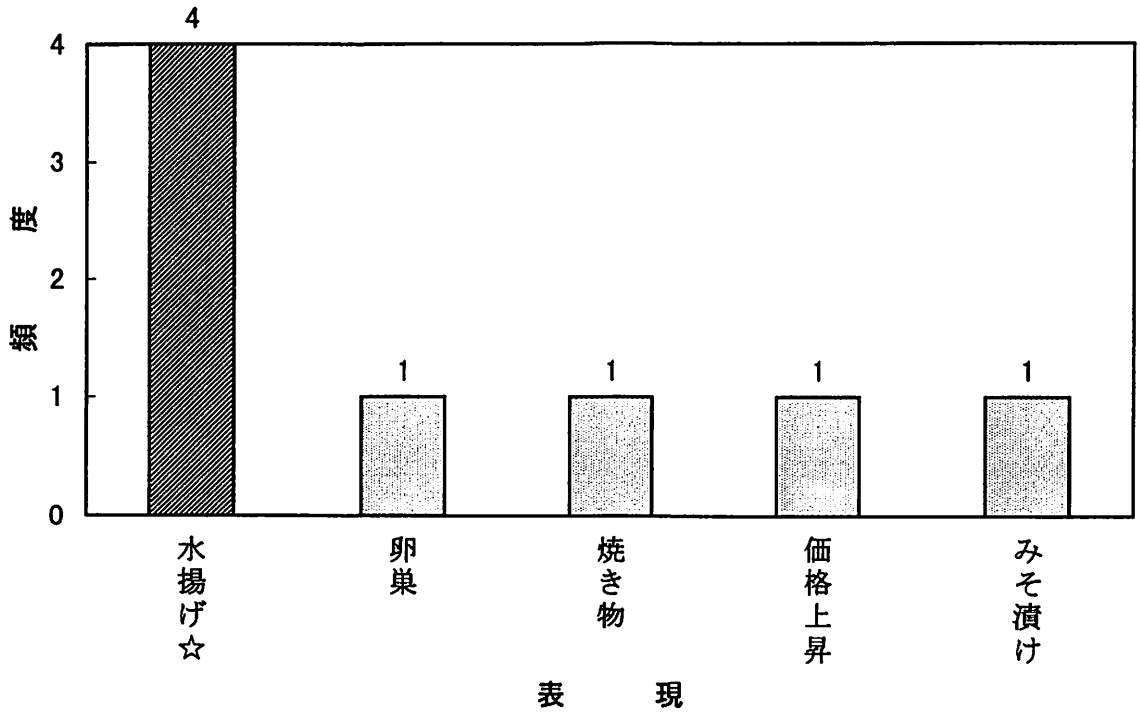
Appendix C

誤抽出の表現の分析グラフ

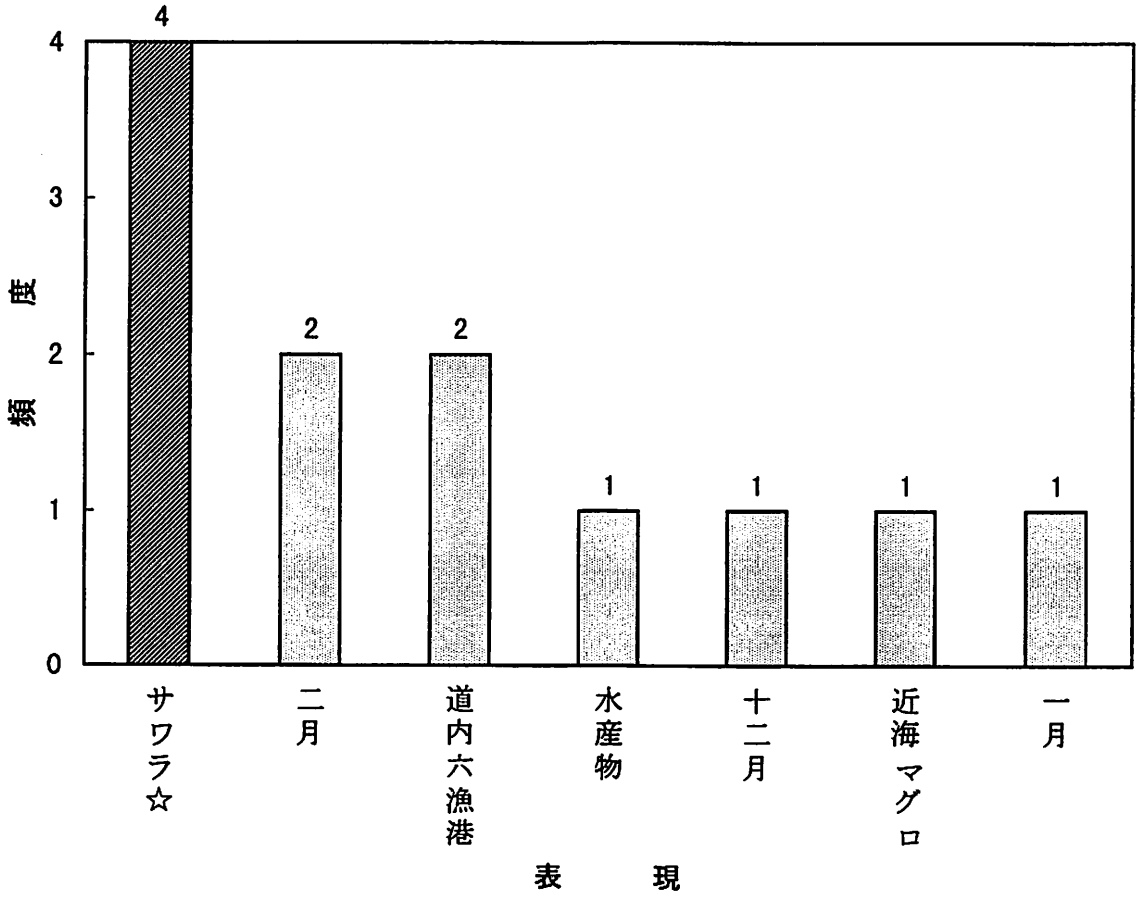
誤抽出(1)

「サワラの水揚げ」

〔サ　ワ　ラ　の　～〕



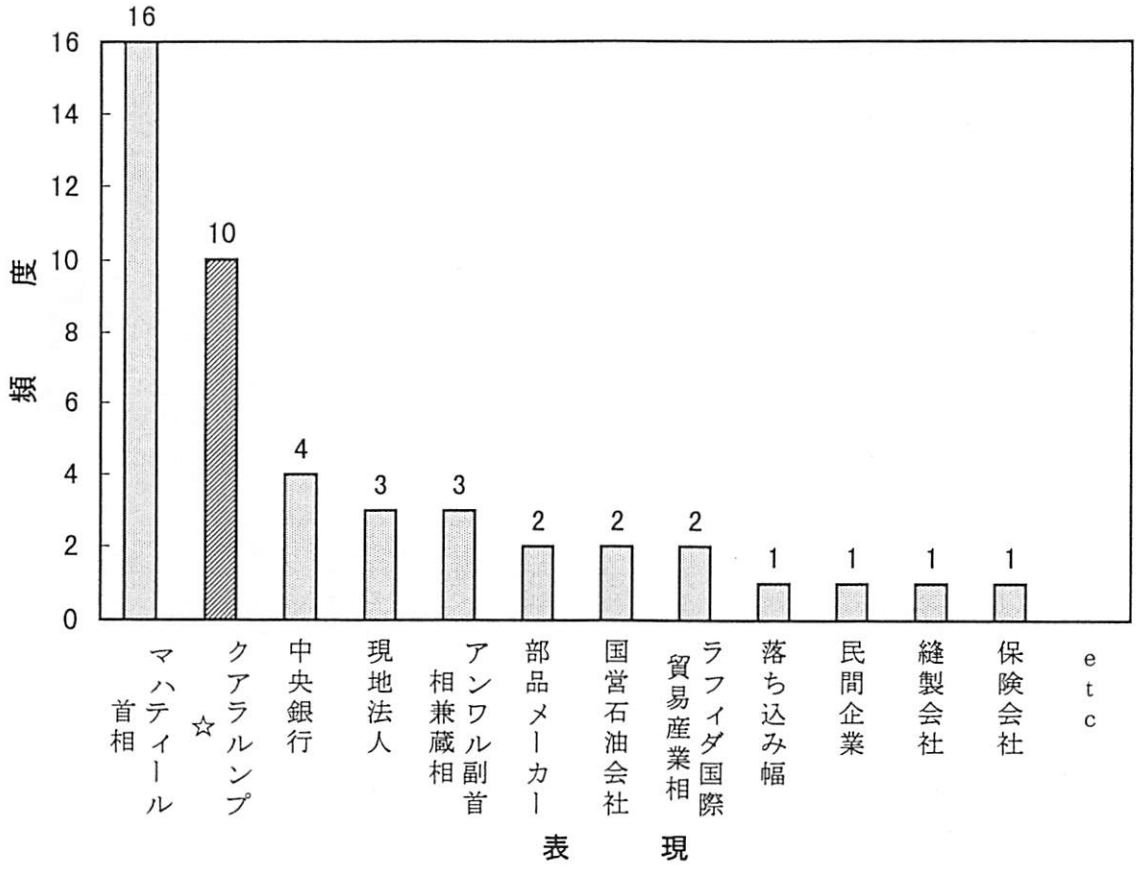
〔～　の　水　揚　げ〕



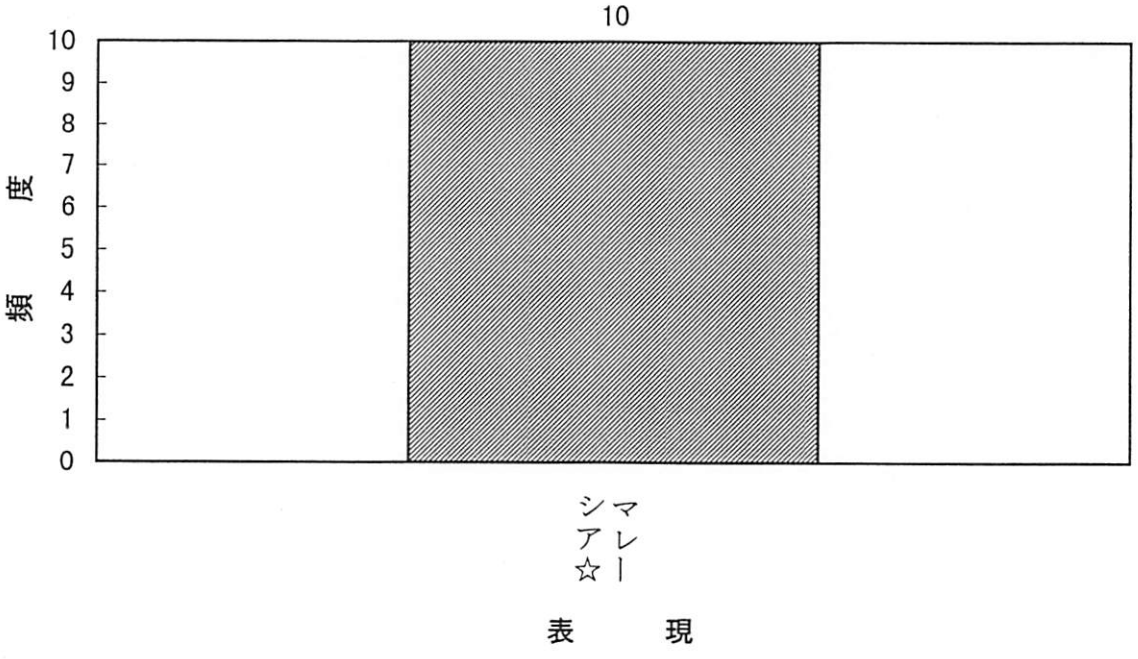
誤抽出(2)

〔マレーシアのクアラルンプ〕

〔マ　レ　ー　シ　ア　の　～〕



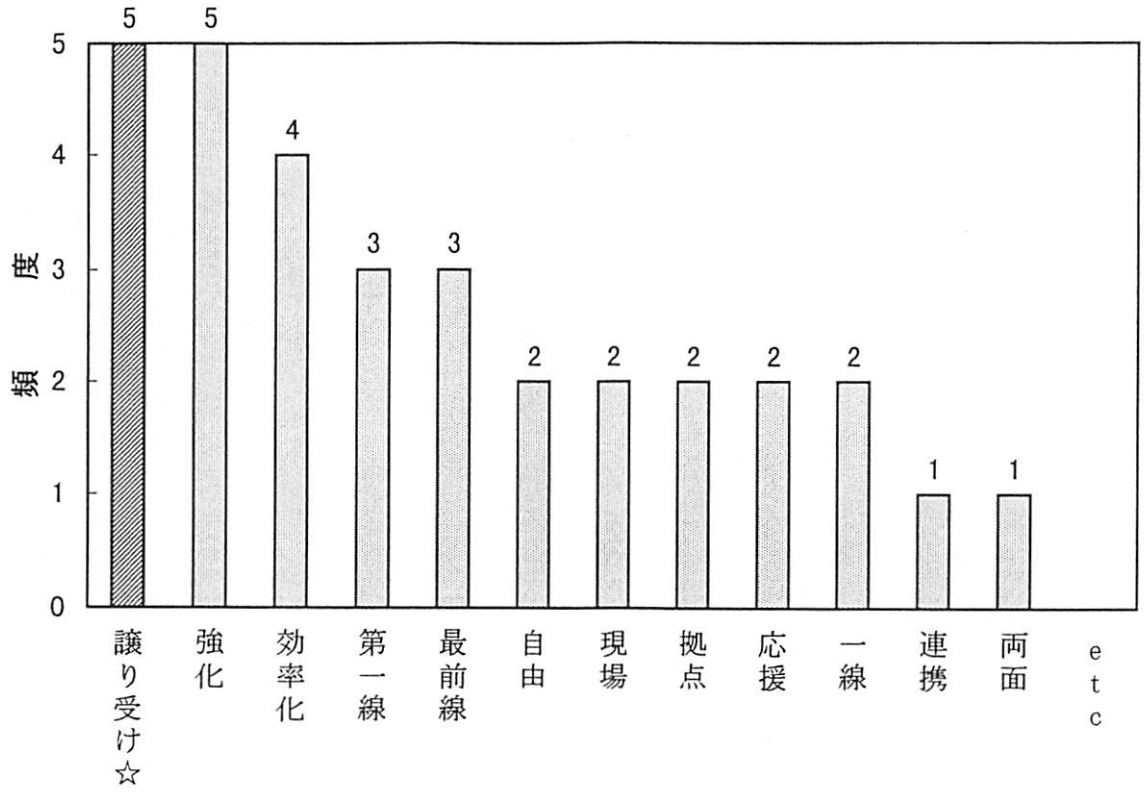
〔～　の　ク　ア　ラ　ル　ン　プ〕



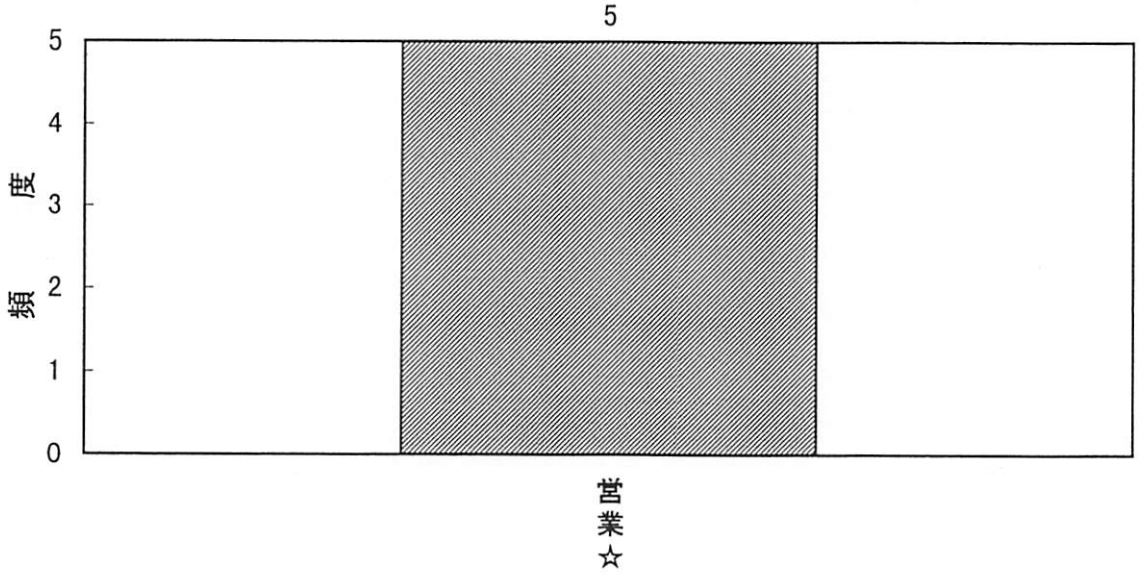
誤抽出(3)

〔営業の譲り受け〕

〔営　業　の　～〕



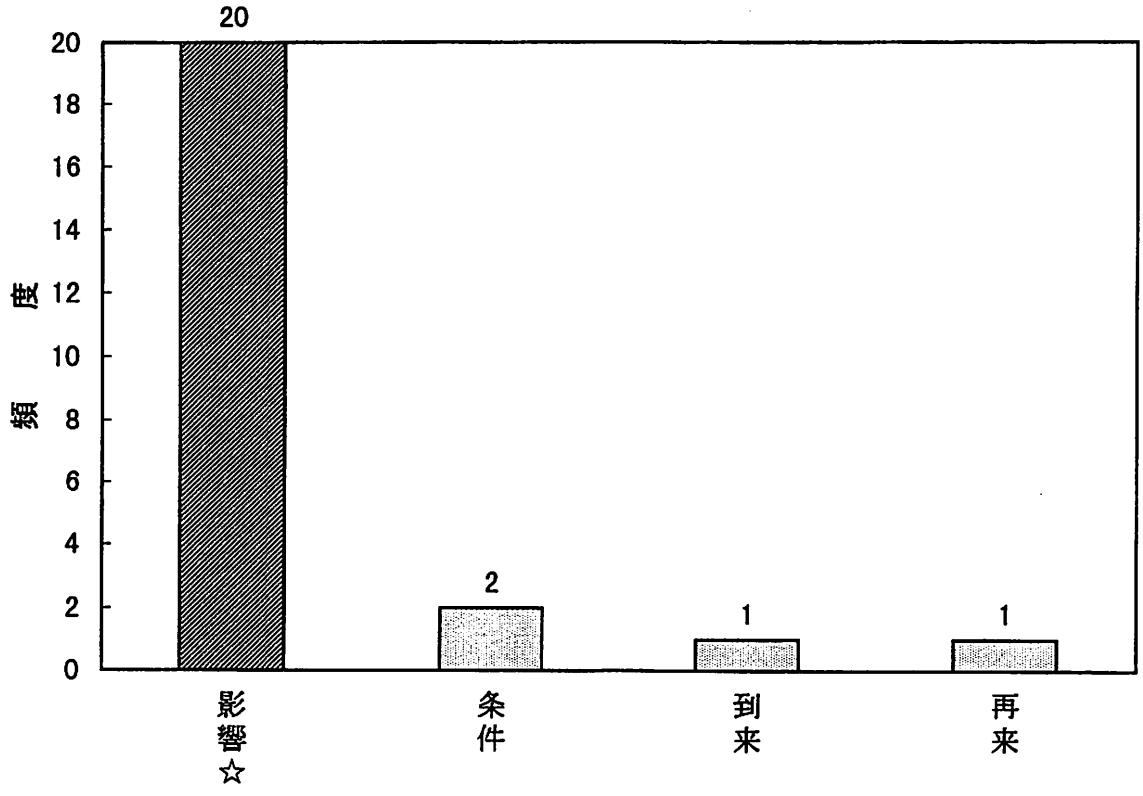
〔～　の　譲　り　受　け〕



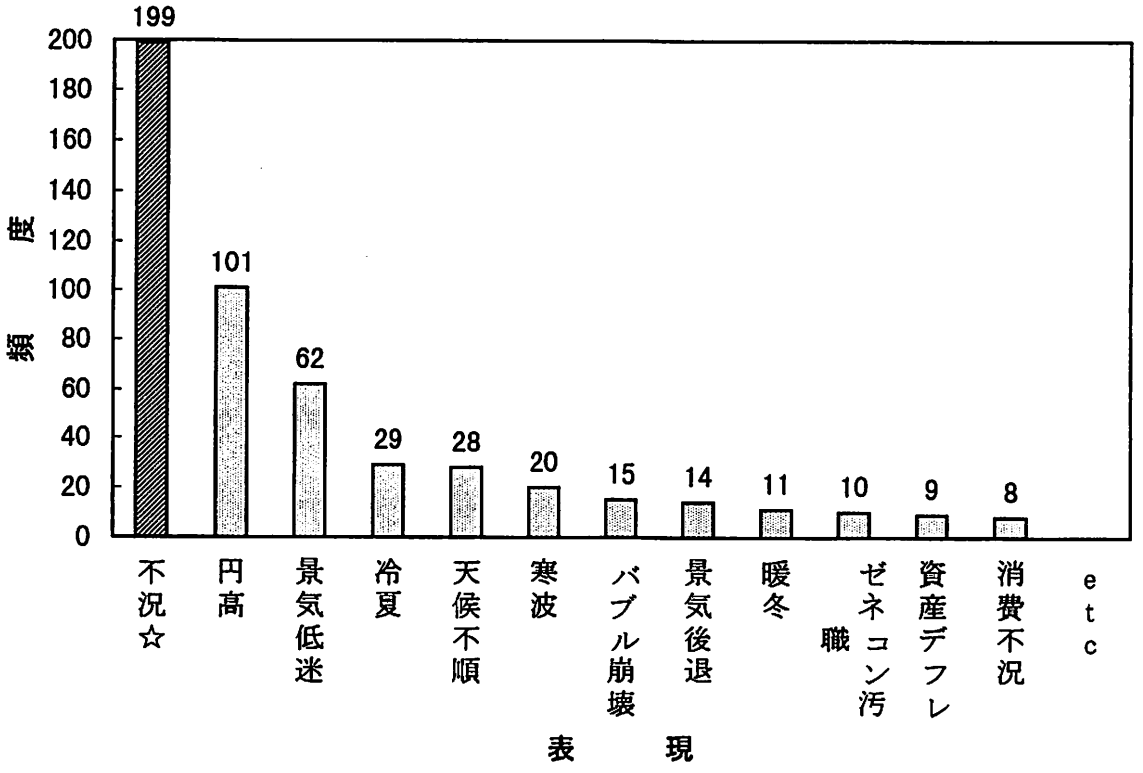
誤抽出(4)

〔寒波の影響〕

〔寒　波　の　～〕



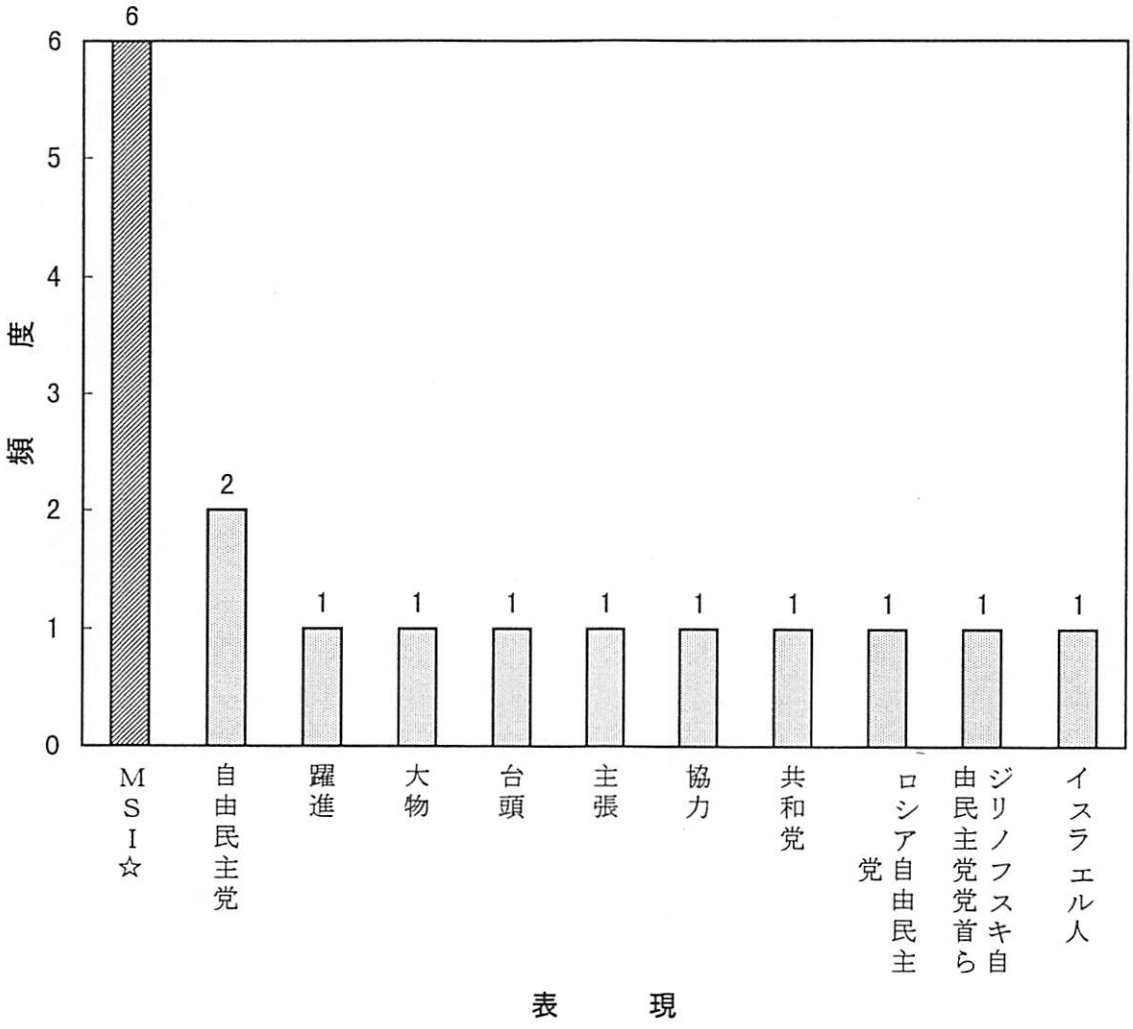
〔～　の　影　響〕



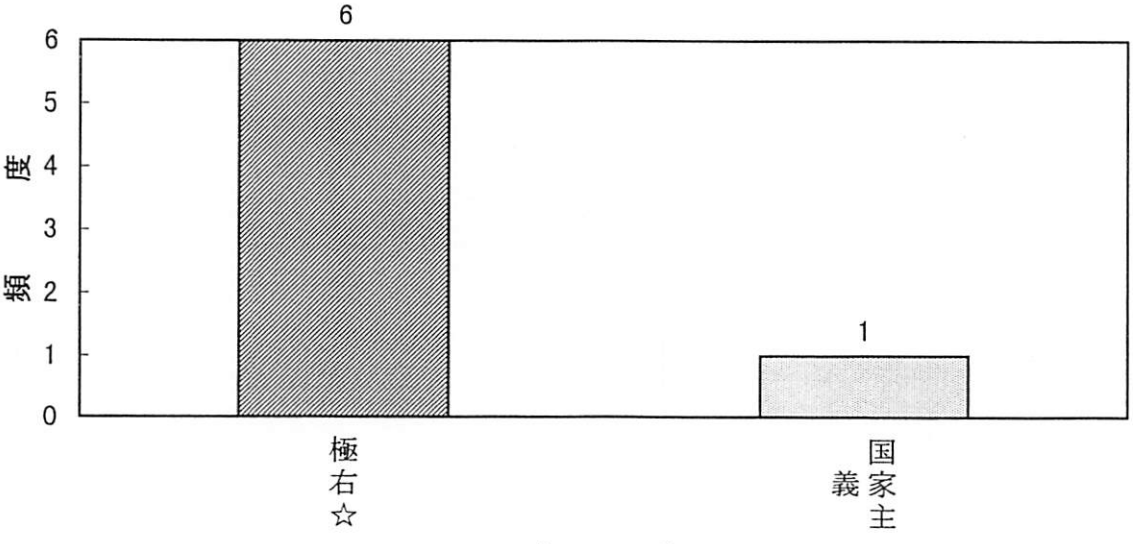
誤抽出(5)

〔極右のMSI〕

〔極　右　の　～〕



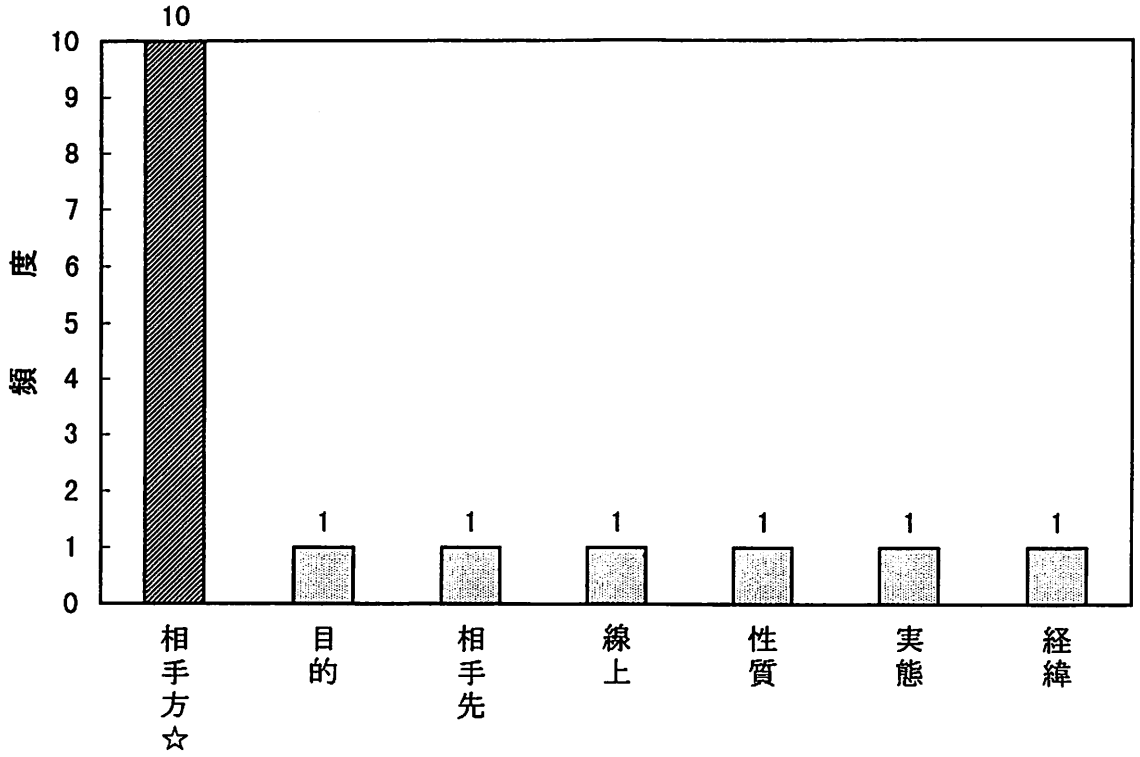
〔～　の　M　S　I〕



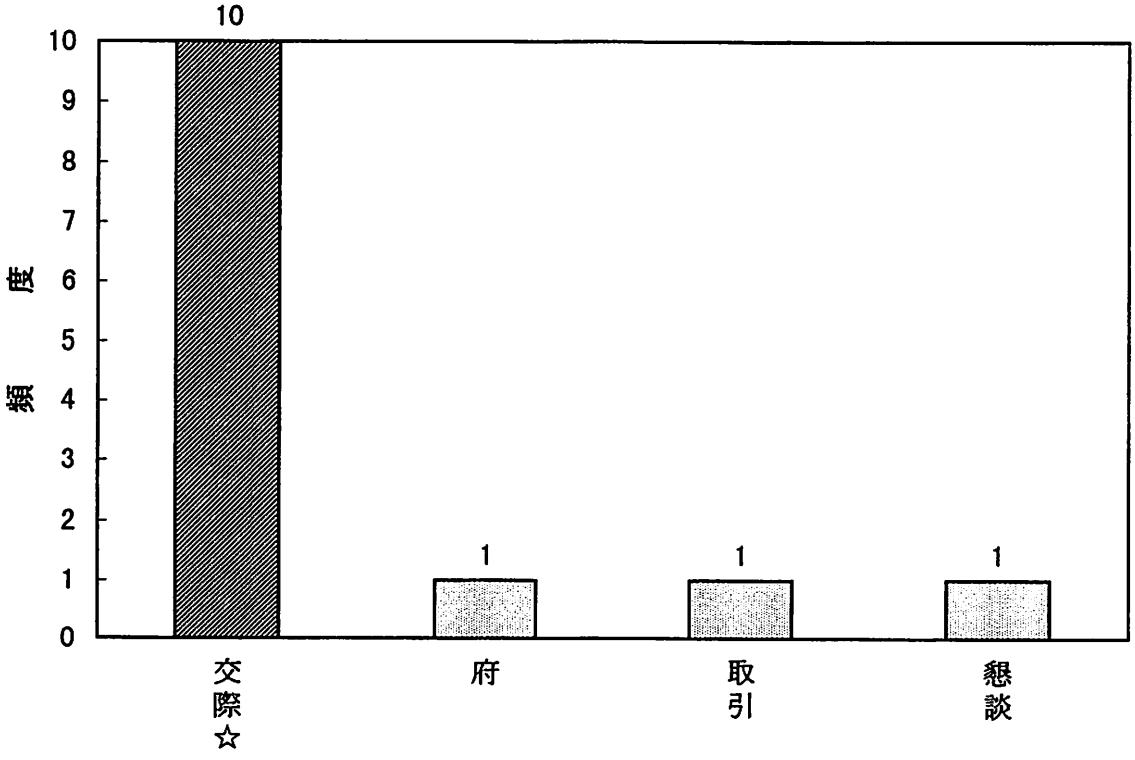
誤抽出(6)

〔交際の相手方〕

〔交　際　の　～〕



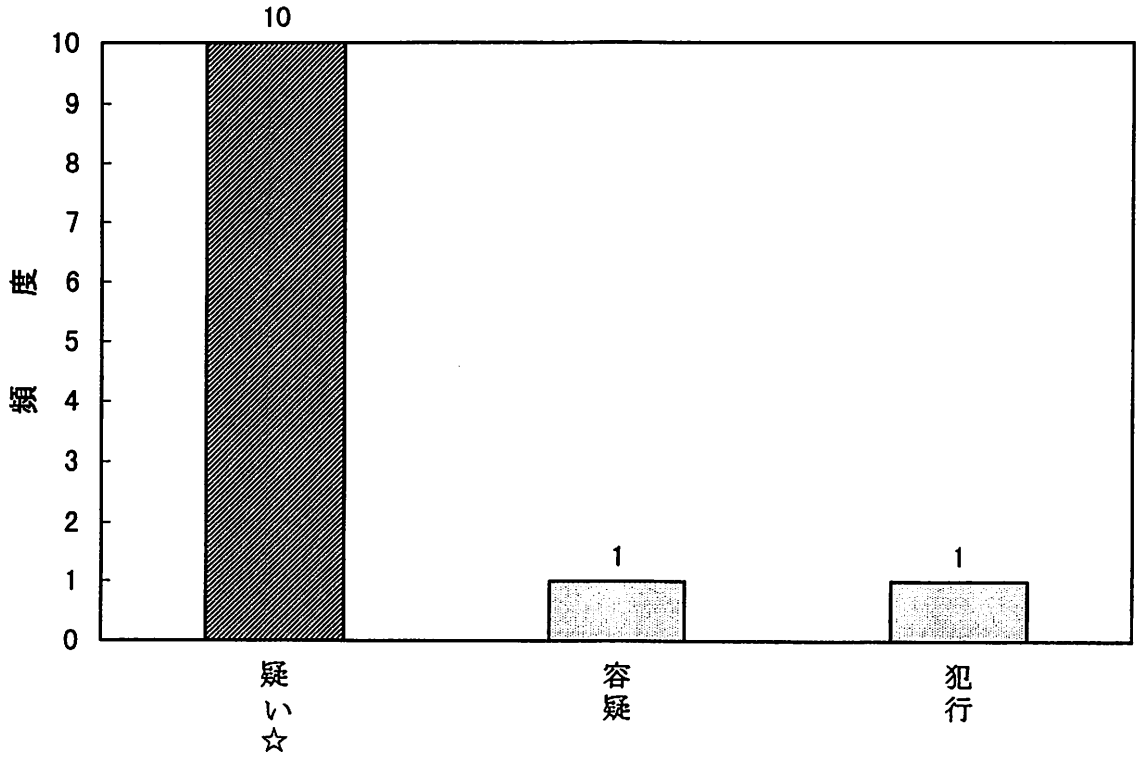
〔～　の　相　手　方〕



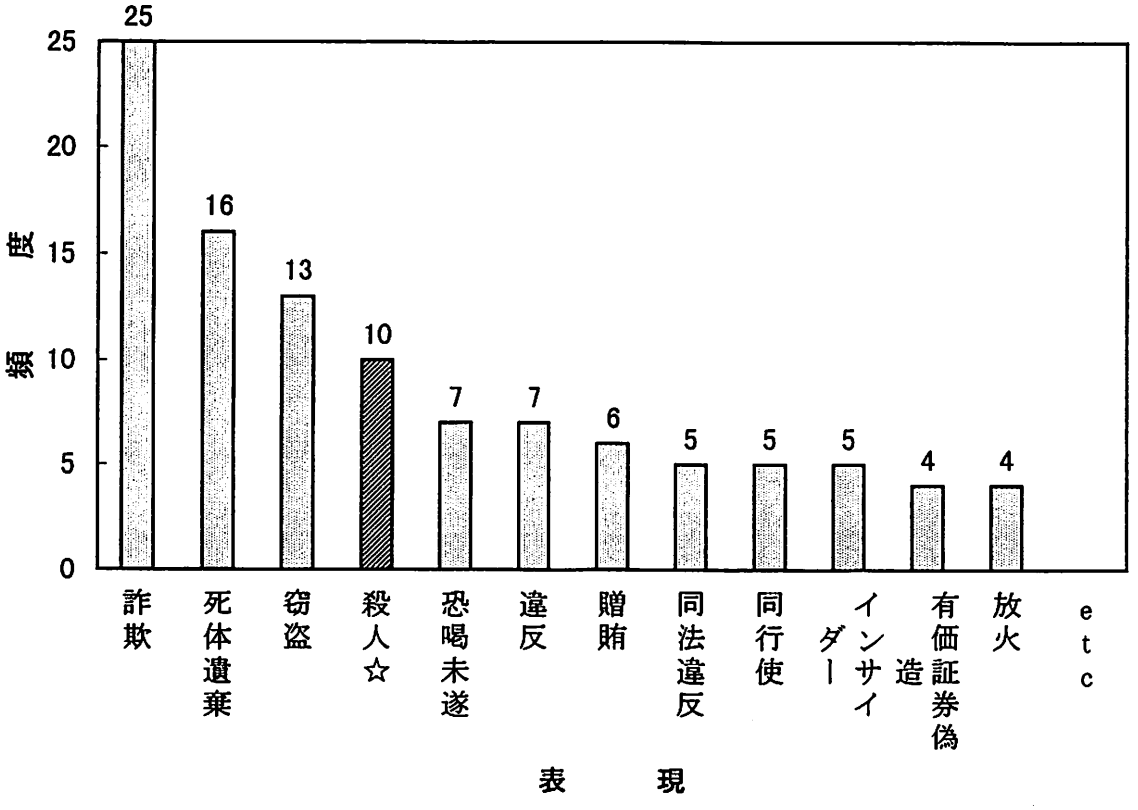
誤抽出(7)

「殺人の疑い」

〔殺　人　の　～〕



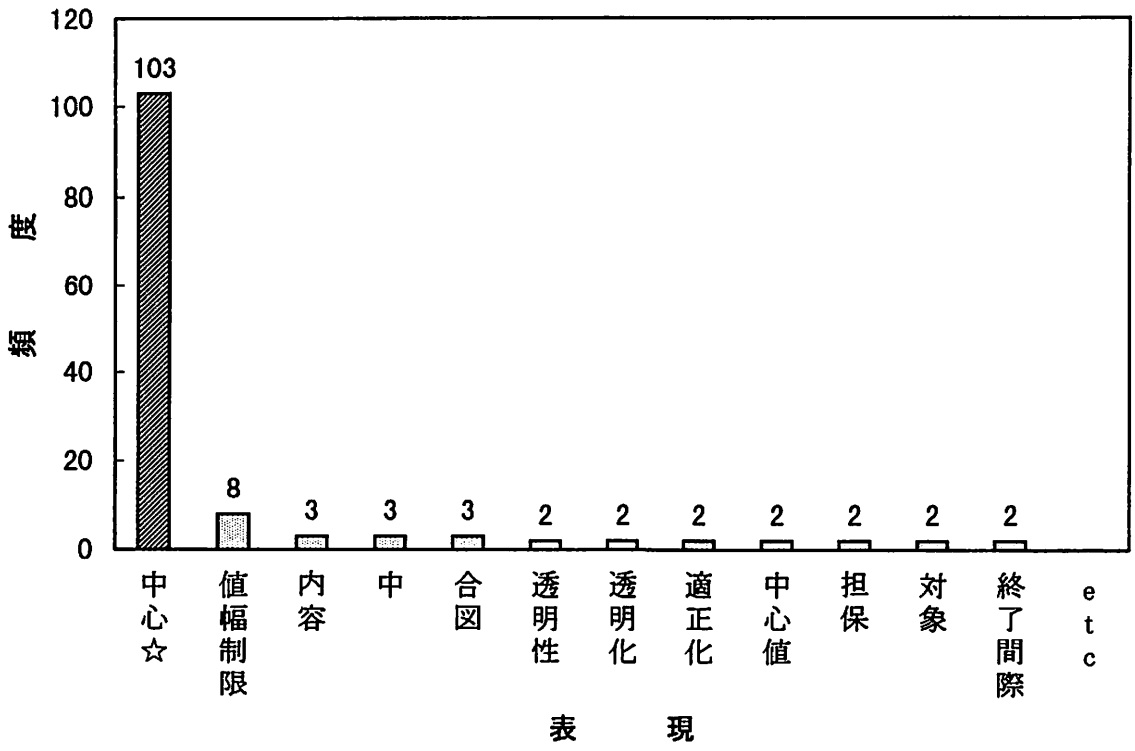
〔～　の　疑　い〕



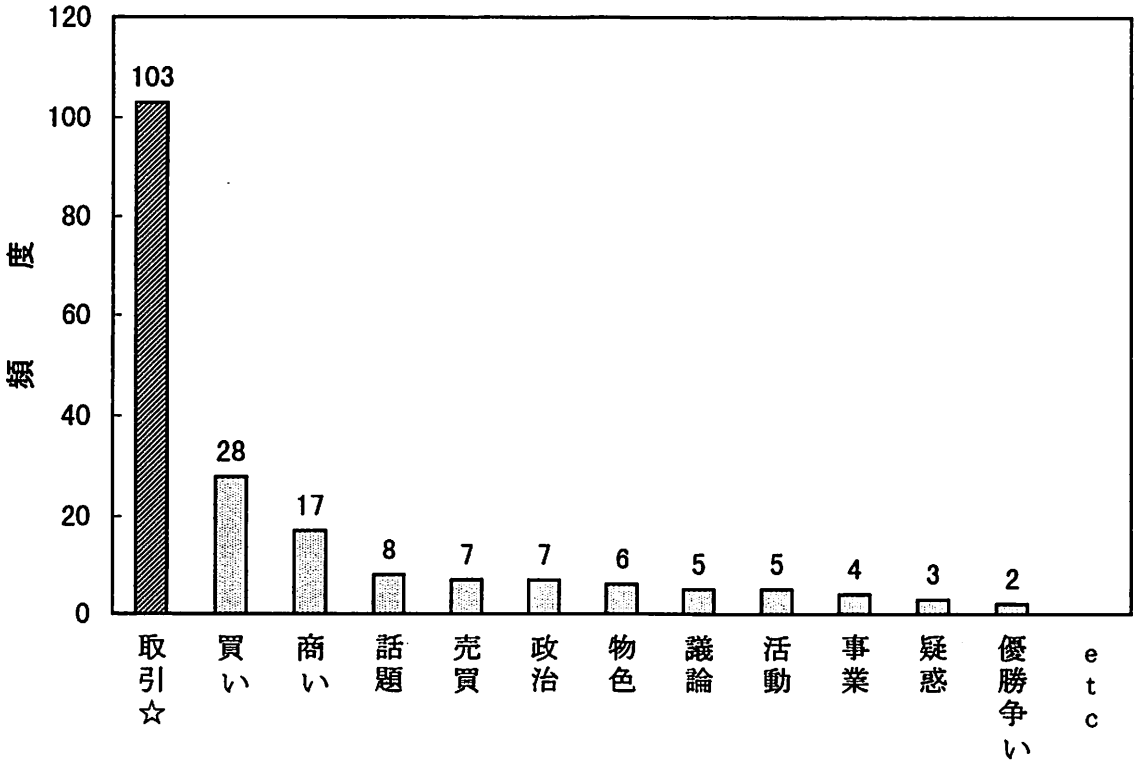
誤抽出(8)

〔取引の中心〕

〔取　引　の　～〕



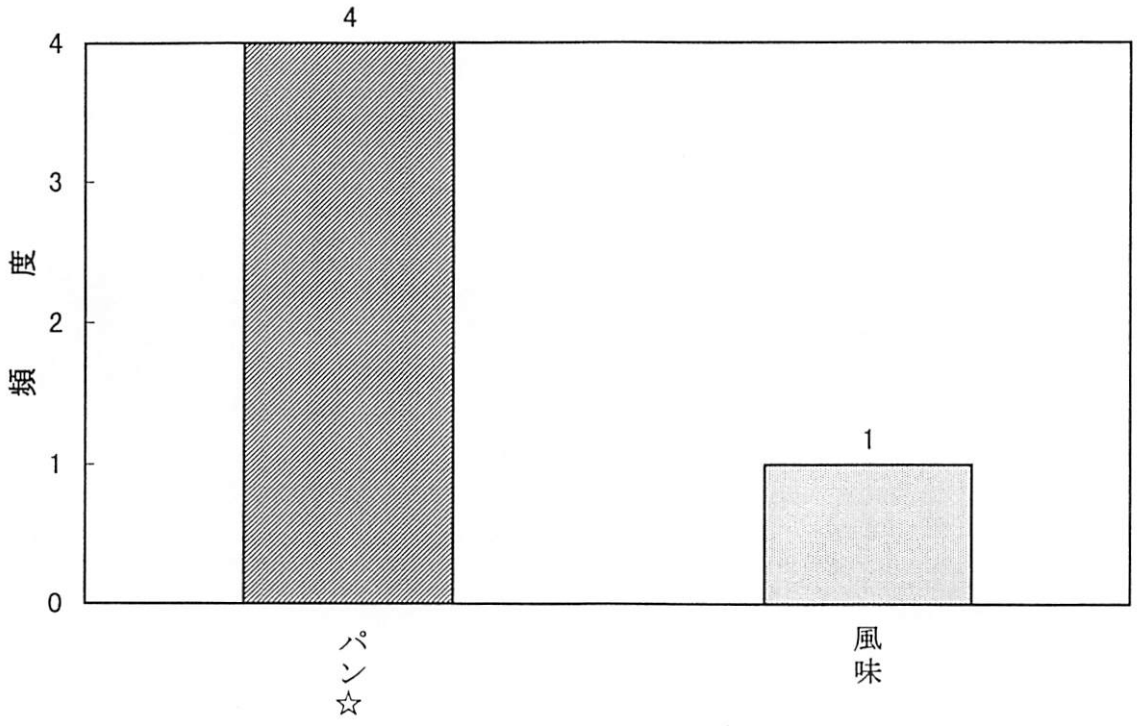
〔～　の　中　心〕



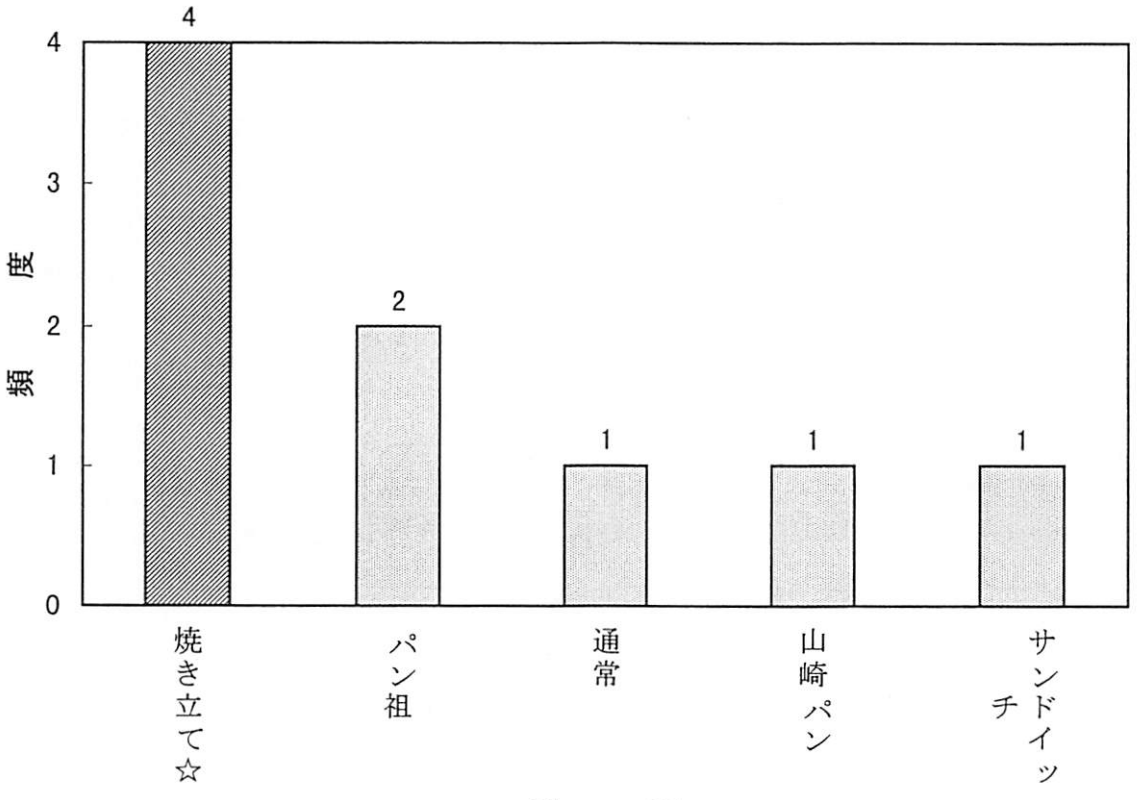
誤抽出(9)

〔焼き立てのパン〕

〔焼　き　立　て　の　～〕



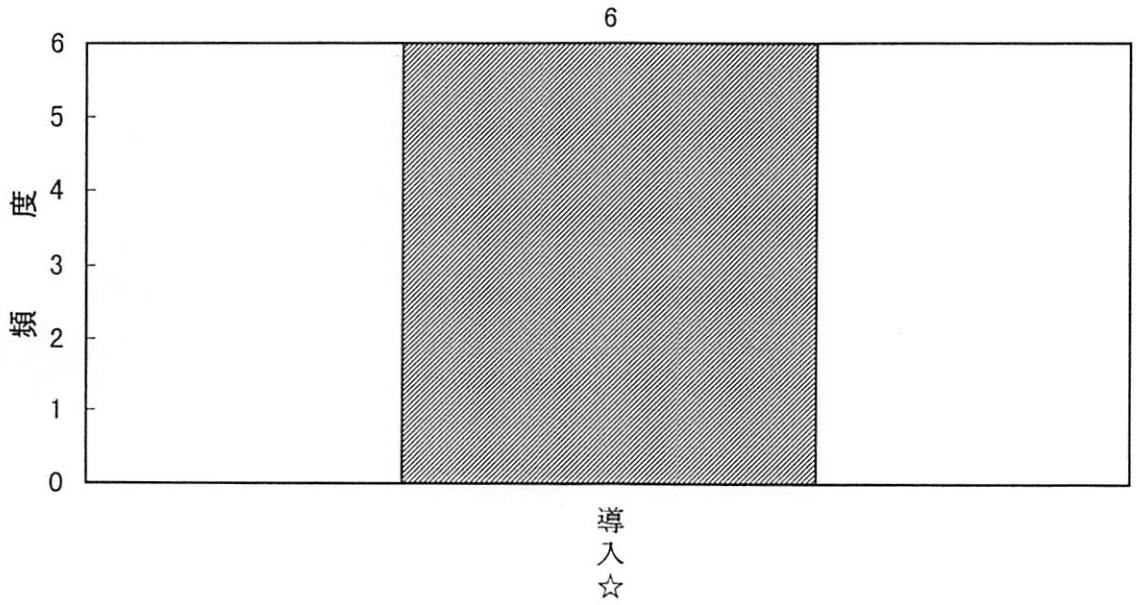
〔～　の　パ　ン〕



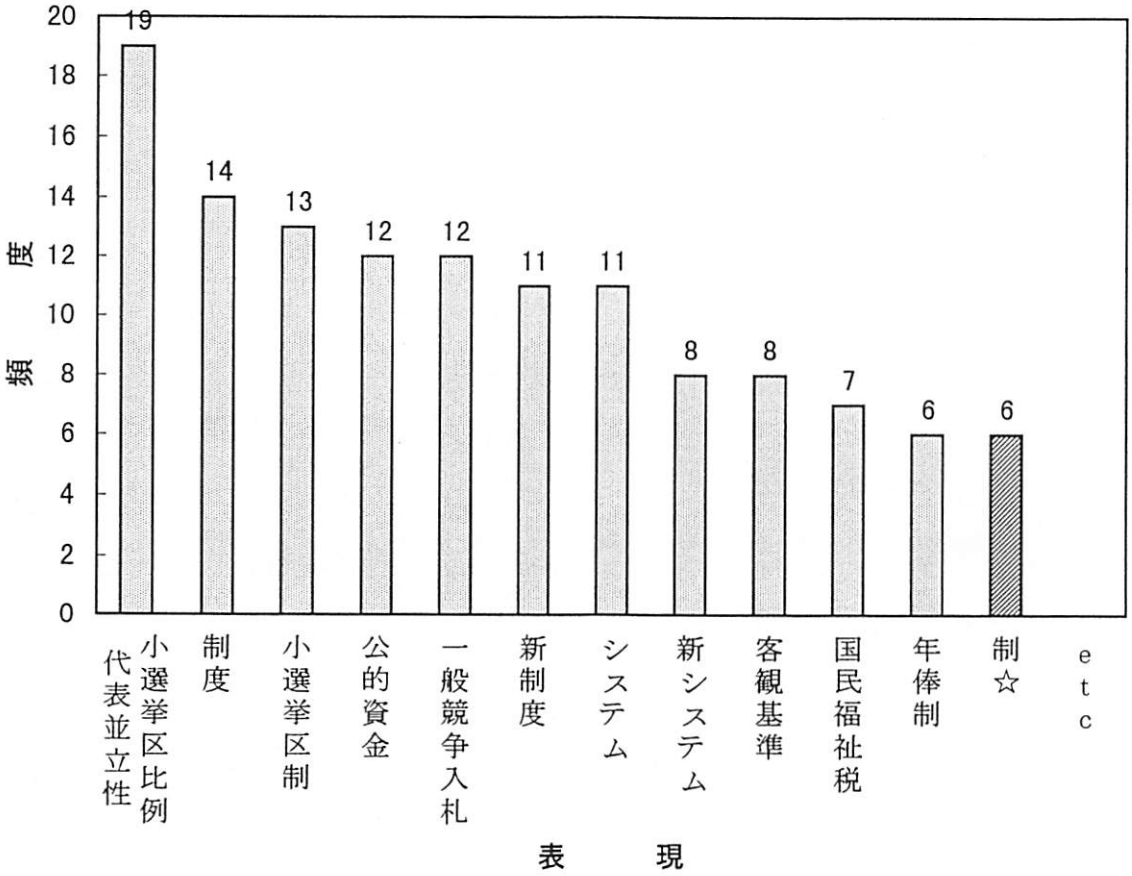
誤抽出(10)

〔制の導入〕

〔制　の　～〕



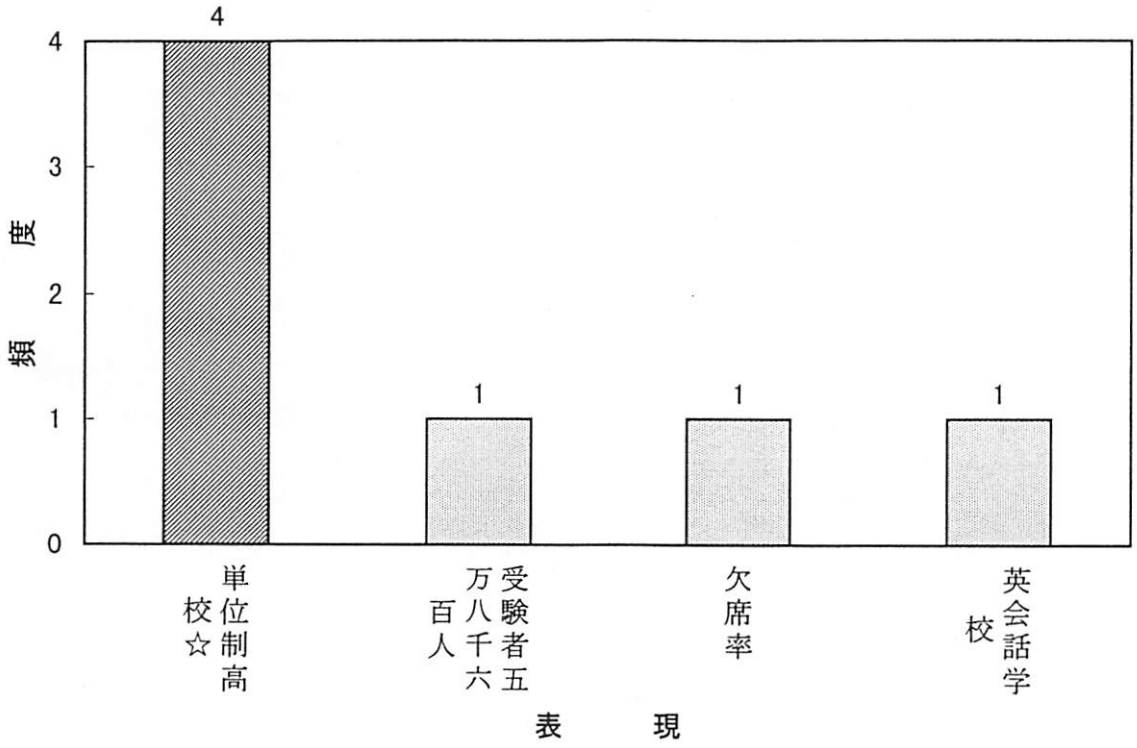
〔～　の　導　入〕



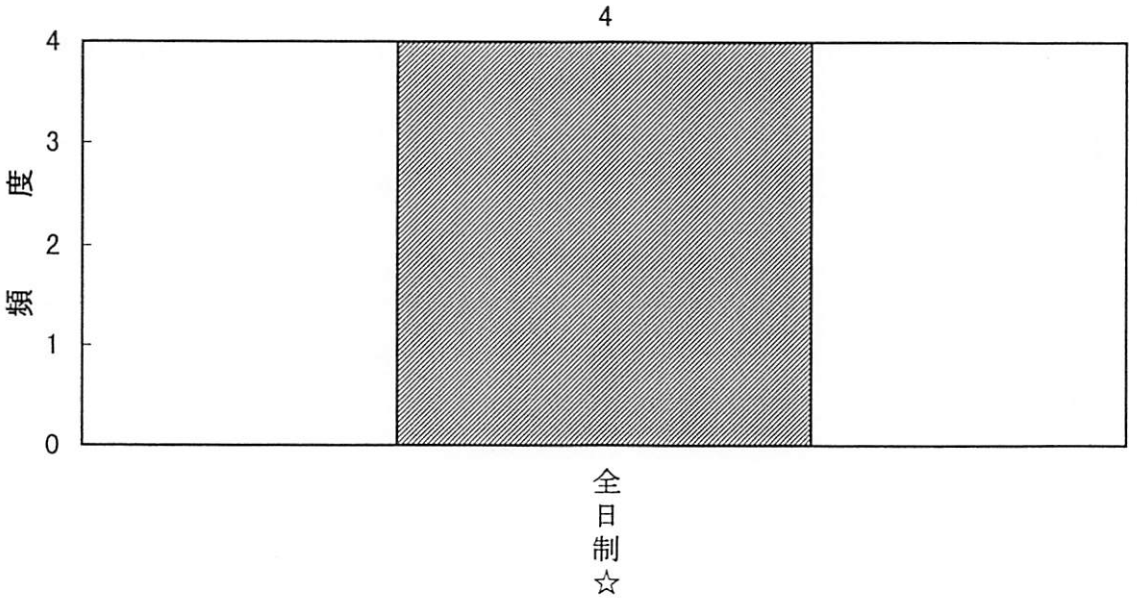
誤抽出(11)

「全日制の単位制高校」

〔全　日　制　の　～〕



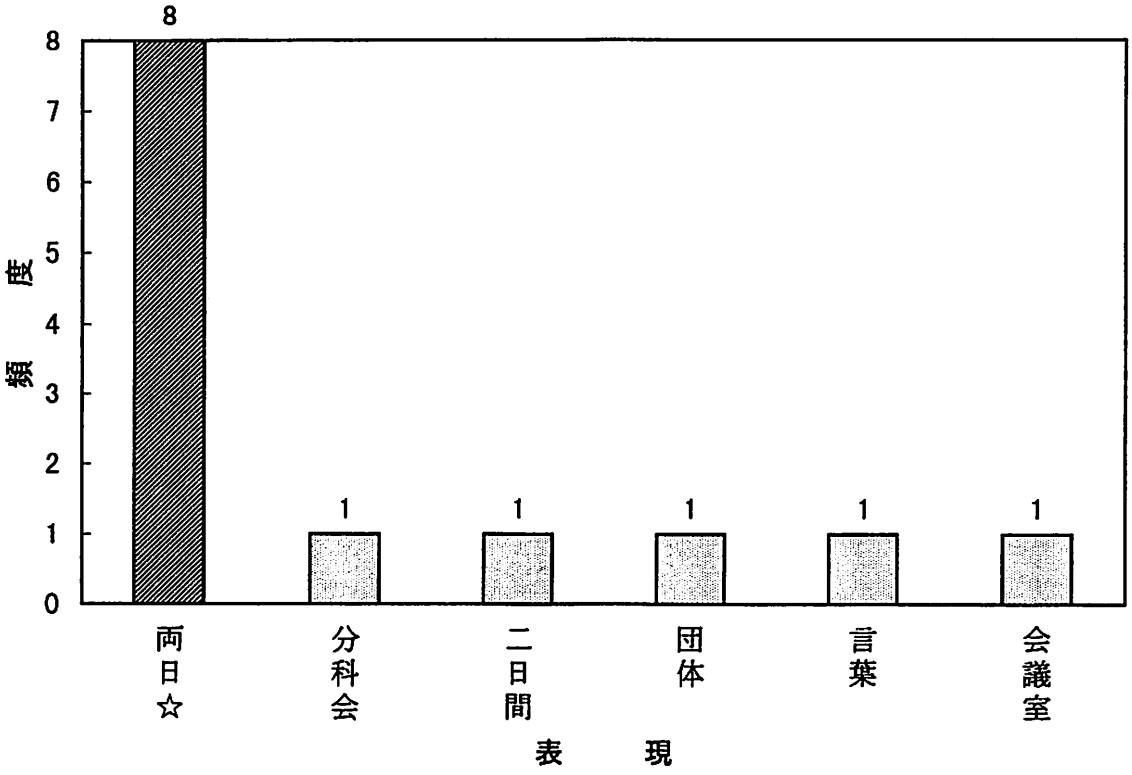
〔～　の　単　位　制　高　校〕



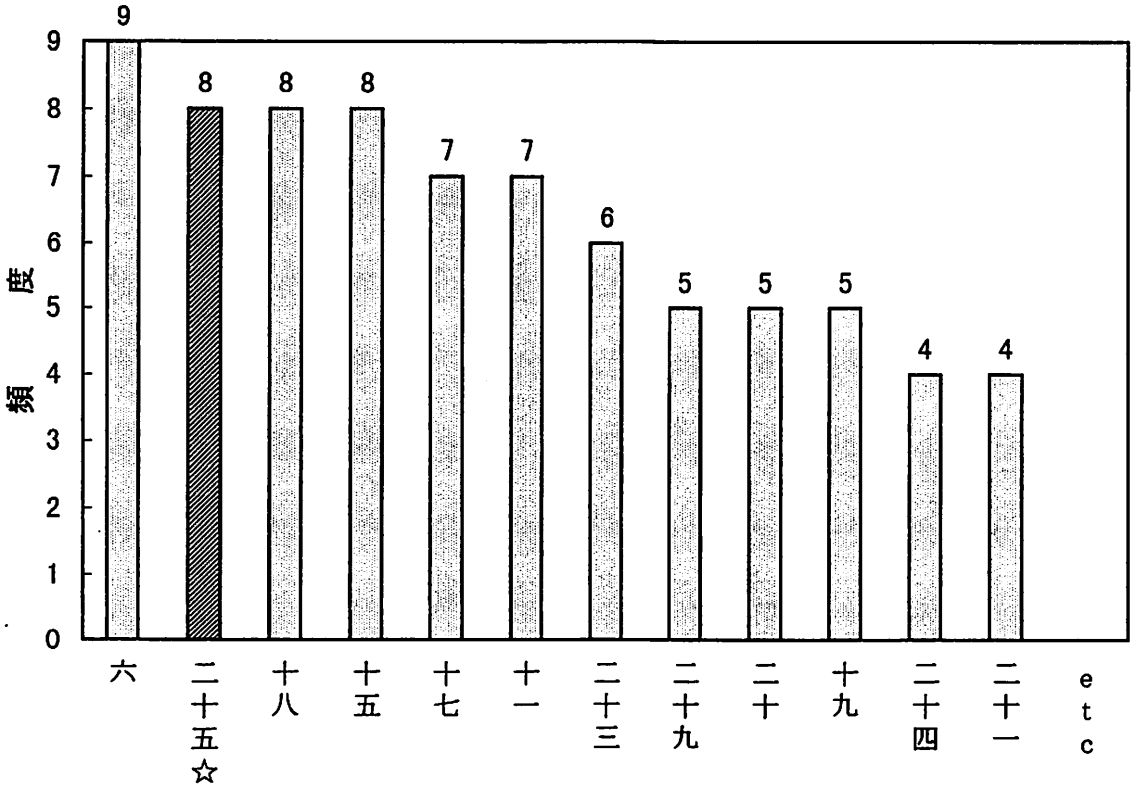
誤抽出(12)

〔二十五の両日〕

〔二　十　五　の　～〕



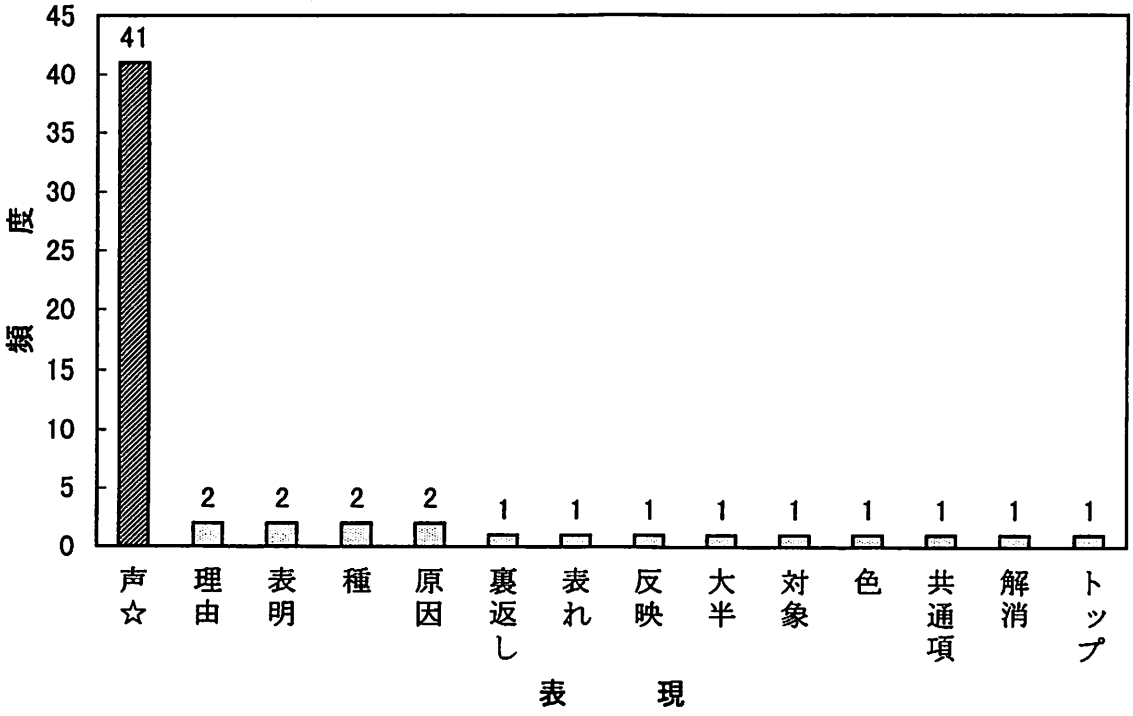
〔～　の　両　日〕



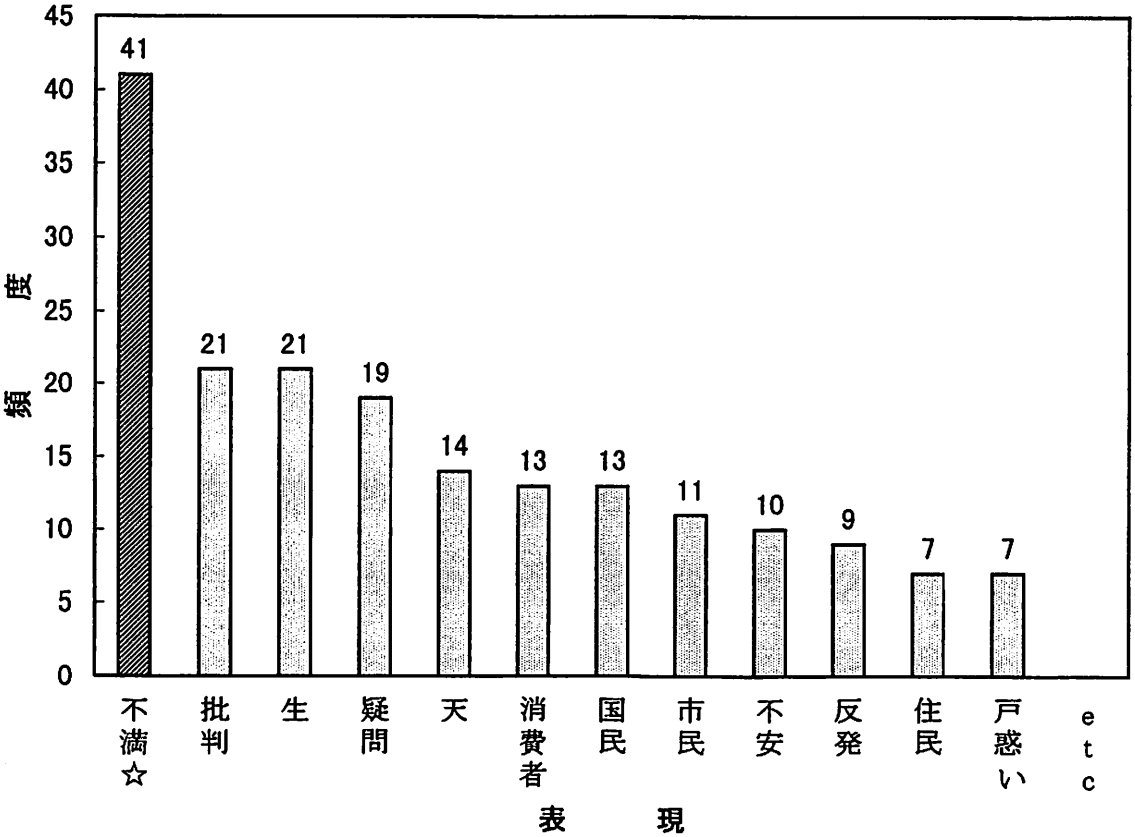
誤抽出(13)

「不満の声」

〔不　満　の　～〕



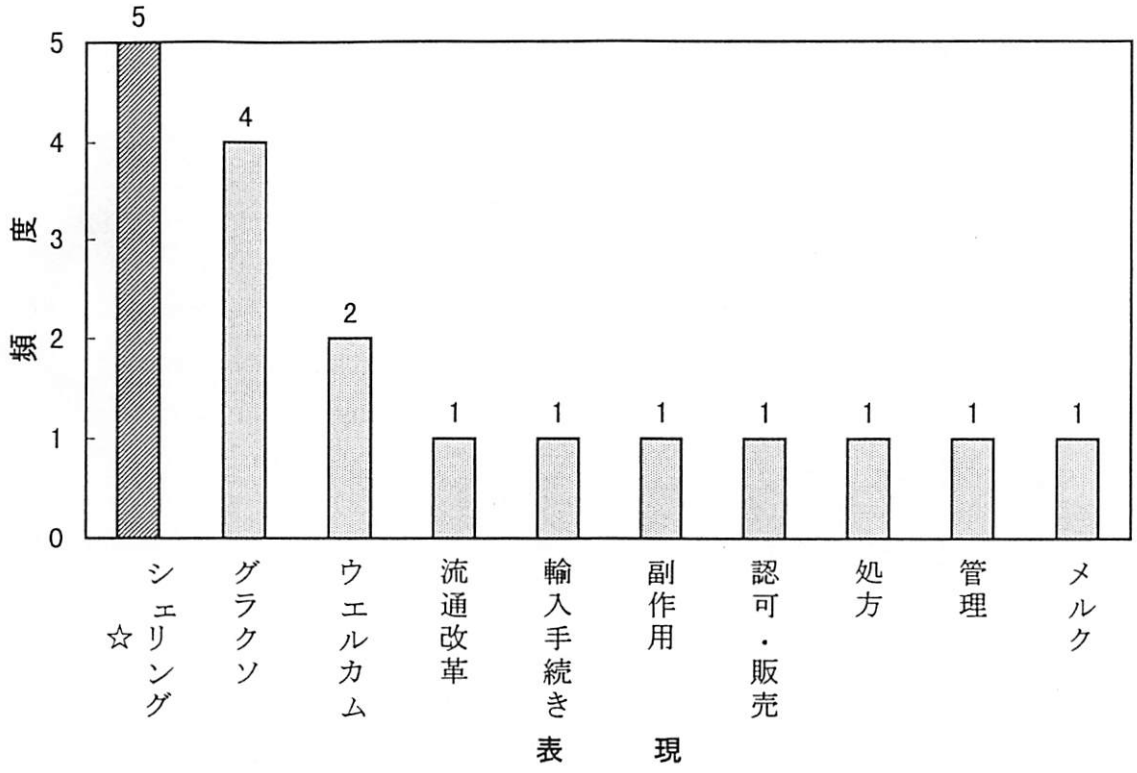
〔～　の　声〕



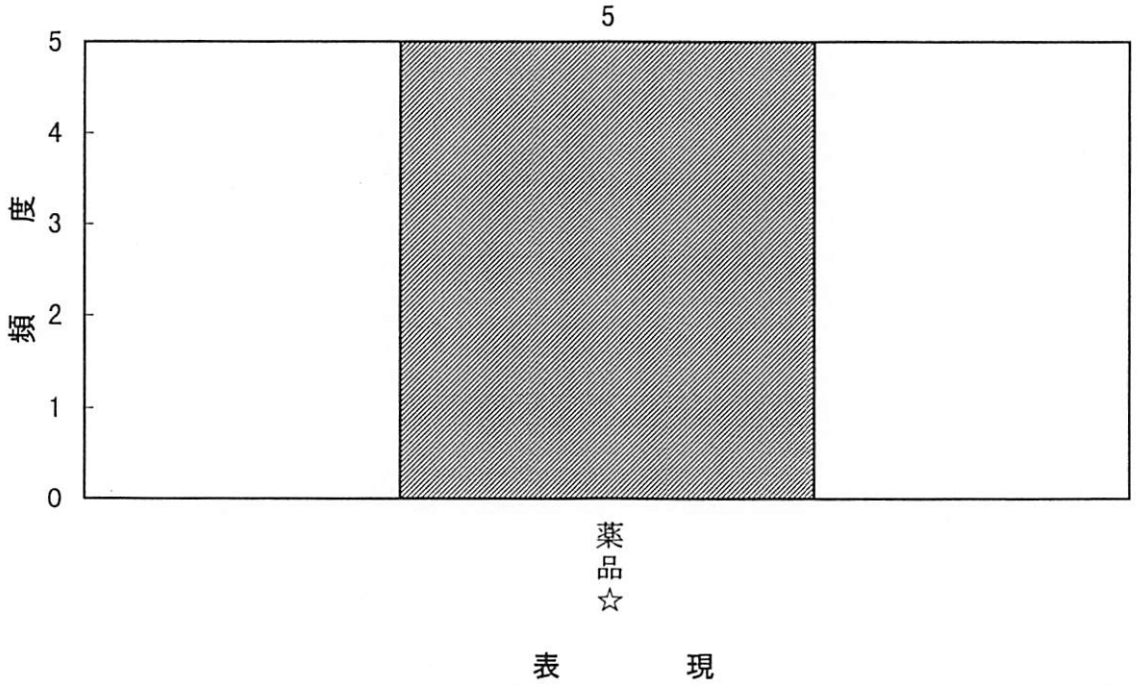
誤抽出(14)

〔薬品のシェリング〕

〔薬　品　の　～〕



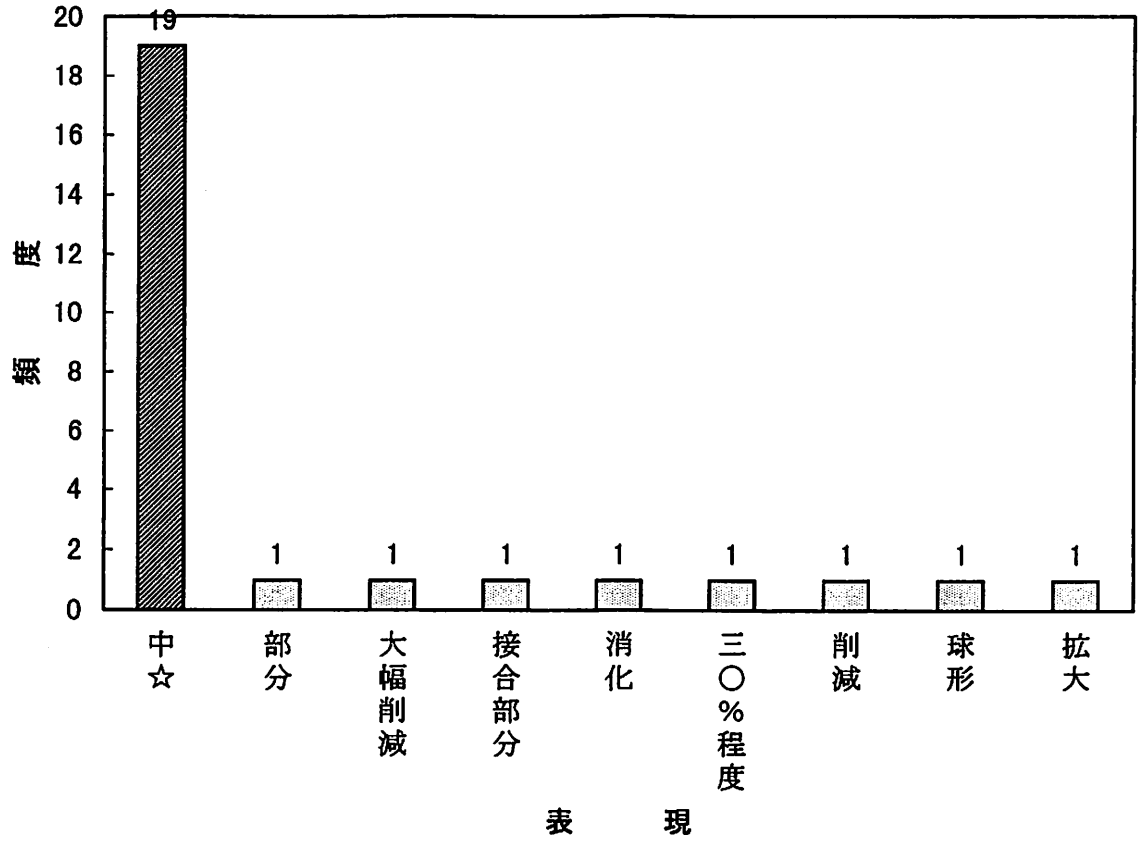
〔～　の　シ　ェ　リ　ン　グ〕



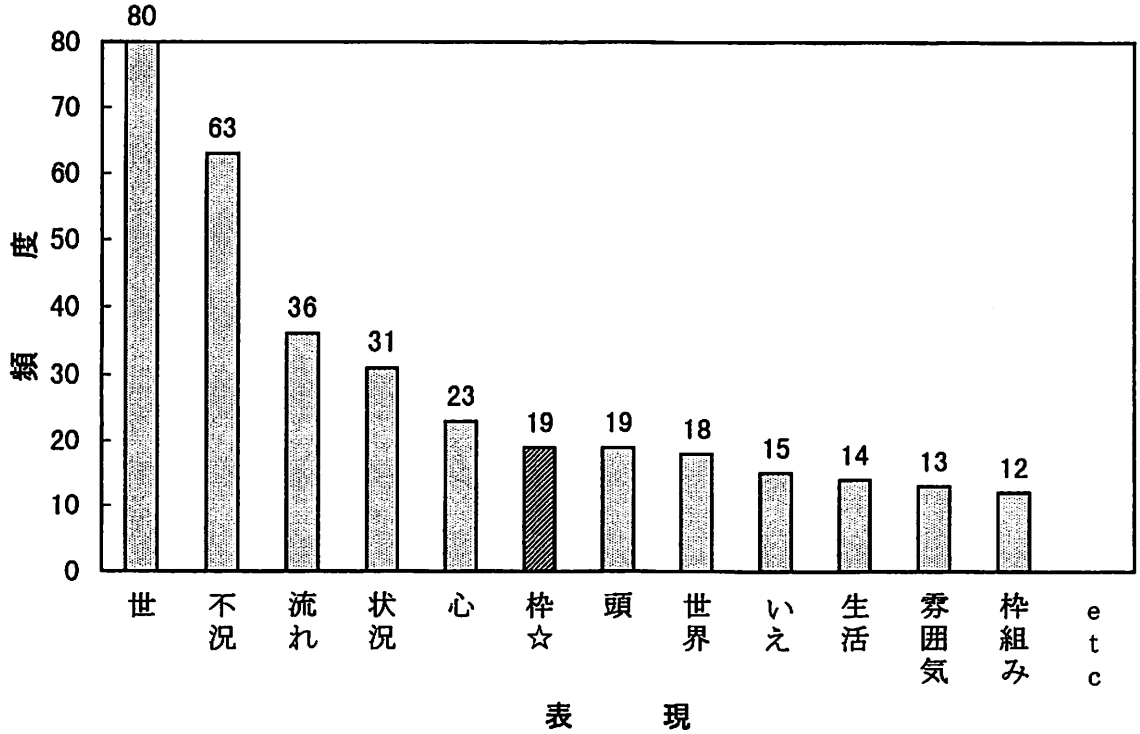
誤抽出(15)

〔枠の中〕

〔枠　の　～〕



〔～　の　中〕



Bibliography

[1] 首藤公昭：“日本語における固定的複合表現”，昭和 63 年度文部省科学研究費補助金特定研究 (I)，課題番号 63101005 (1989).

[2] 新納浩幸, 井佐原均：“片方向の共起性による述語型定型表現の自動抽出”，言語処理学会、Vol.2, No.3, pp.73-86 (1995).

[3] Sophia Ananiadow：“Towards a Linguistic Treatment of Compounds in a Machine Translation Environment”, Journal of Natural Language Processing, Vol.3, No.1, pp.45-66 (1996).

[4] 菅民郎：“多変量解析の実践”，現代数学社 (1993).

[5] 松本裕治, 黒橋禎夫, 宇津呂武仁, 妙木裕, 長尾真：“日本語形態素解析システム juman 使用説明書 ver 2.0”，京都大学長尾研究室 (1994).

[6] J.R. キンラン＝著 古川康一＝監訳：“AI によるデータ解析”，トッパン (1993).

[7] Kenji KITA et al：“Application of Corpora in Second Language Learning” - The Problem of Collocational Knowledge Acquisition -” Second Annual Workshop on Very Large Corpora (WVLC2),pp.43-56 (1994)

[8] Church,K.W. and Hanks,P.：“Word association norms,mutual information,and lexicography” Computational Linguistics,Vol.16,No.1,pp.22-29 (1990)

[9] 松本裕治：“コーパスに基づく自然言語処理”，電子情報通信学会主催『自然言語処理におけるコーパスの利用』講習会テキスト (1996).