(19)日本国特許庁（ＪＰ）　(12) **公開特許公報**（Ａ）　(11)特許出願公開番号
特開2003－58860
(P2003－58860A)
(43)公開日　平成15年２月28日(2003.2.28)

(51)Int.Cl.[7]	識別記号	ＦＩ	テーマコード(参考)
Ｇ０６Ｎ　3/00	５５０	Ｇ０６Ｎ　3/00	５５０Ｚ　5Ｂ０９１
Ｇ０６Ｆ　17/27		Ｇ０６Ｆ　17/27	Ｘ

審査請求　有　　請求項の数４　ＯＬ　（全 6 頁）

(21)出願番号　特願2001－246643(P2001－246643)

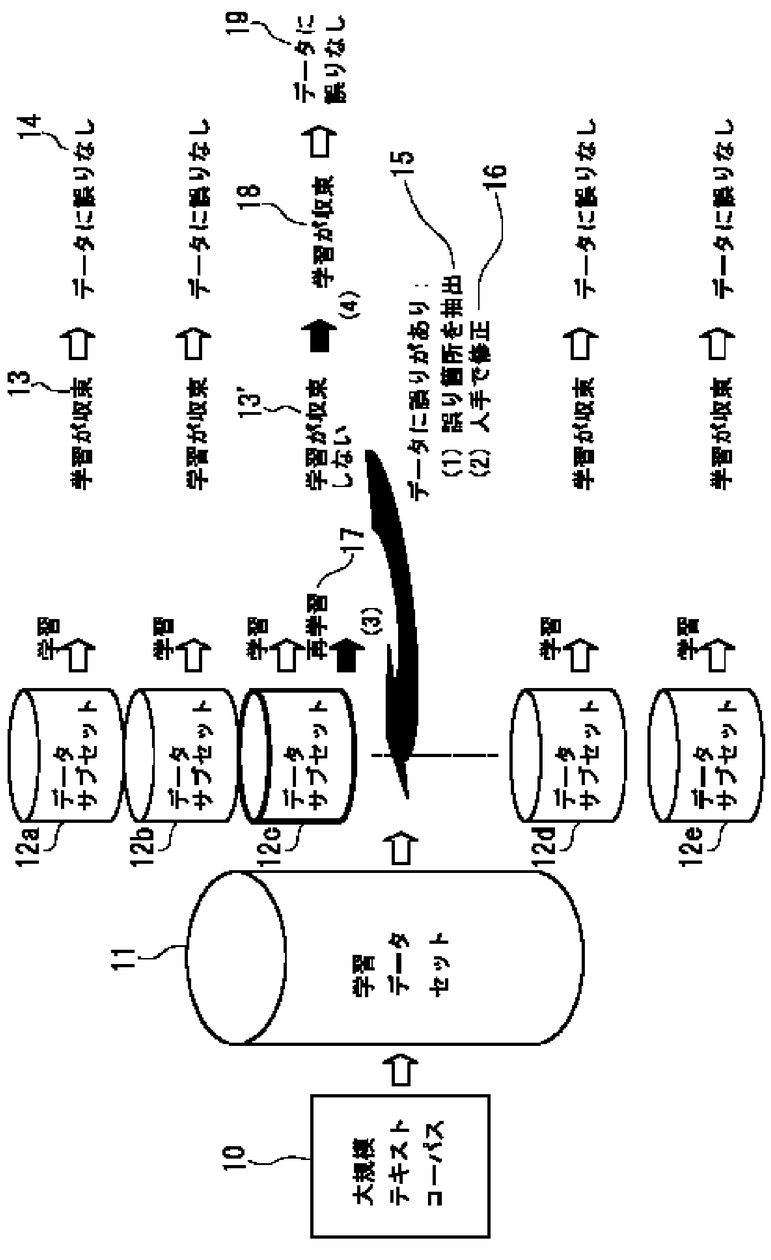
(22)出願日　平成13年８月15日(2001.8.15)

(71)出願人　301022471
独立行政法人通信総合研究所
東京都小金井市貫井北町４－２－１
(72)発明者　馬　青
東京都小金井市貫井北町４－２－１　独立行政法人通信総合研究所内
(74)代理人　100090893
弁理士　渡邊　敏

Ｆターム(参考)　5B091 AA15 BA02 EA01 EA04

(54)【発明の名称】　テキストコーパスの誤り検出方法及び装置

(57)【要約】　（修正有）
【課題】　言語列の作成者や配布者らが、容易に所望の情報を埋め込むことが出来ると同時に、読者や利用者がその存在に気付き難く、しかも所定の方式に基づけば確実に抽出が可能な情報埋込・抽出方法及びその装置並びに記録媒体を提供すること。
【解決手段】　言語列における各行の終端近傍の文字単位を構成する行末文字群と改行位置との位置関係を変化させることによって有意な情報を埋め込む。前記言語列には、実体的に印刷・表示可能な言語列や、電磁的に記録された言語列を対象とすることもできる。文字単位が形態素による区分でもよい。これらの情報埋込方法によって作成・出力された言語列から情報抽出する方法を提供する。そして、情報埋込・抽出を行う装置や、言語列を記録する記録媒体を提供する。



【特許請求の範囲】
【請求項１】単語情報を含む予め作成されたテキストコーパスにおける該単語情報の誤りを検出する方法であって、
該各単語情報の分類をニューラルネットワークにおけるクラスとして捉え、
それらを小規模な２クラス問題に分割して、複数のモジュールを構成し、
各モジュールがニューラルネットワークにおける学習過程において収束するか否かの演算を行い、
収束しない場合に、該モジュールに該単語情報の誤りがあると判定し、該モジュールを抽出することを特徴とするテキストコーパスの誤り検出方法。
【請求項２】前記単語情報が、品詞に係る情報であって、
該情報をタグ形式でテキスト中に埋め込み、テキストコーパスを構成し、
該タグの誤りを検出する請求項１に記載のテキストコーパスの誤り検出方法。
【請求項３】単語情報を含む予め作成されたテキストコーパスにおける該単語情報の誤りを検出する検出装置であって、
該各単語情報の分類をニューラルネットワークにおけるクラスとして捉え、
それらを小規模な２クラス問題に分割して、複数のモジュールを構成し、
各モジュールがニューラルネットワークにおける学習過程において収束するか否かの演算を行い、
収束しない場合に、該モジュールに該単語情報の誤りがあると判定し、該モジュールを抽出する一連の処理を行うことによって誤りを検出可能なことを特徴とする検出装置。
【請求項４】前記単語情報が、品詞に係る情報であって、
該情報をタグ形式でテキスト中に埋め込み、テキストコーパスを構成し、
該タグの誤りを検出する請求項３に記載のテキストコーパスの誤り検出装置。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、言語処理に用いられるテキストコーパスの誤りを検出する方法に関し、より詳しくは、該誤り検出の高速化、高効率化に関する技術である。
【０００２】
【従来の技術】近年、さまざまなテキストコーパスが作られ、教師有り機械学習の研究をはじめとして、言語処理技術の研究が盛んに行われている。しかし、学習に用いられるテキストコーパスは人手によって作成されるため、多くの誤りを含み、この誤りが各研究の進捗を妨げたり、言語処理精度の低下を招く場合も多い。このため、テキストコーパス中の誤りを検出・修正することは非常に重要な課題となっている。
【０００３】従来から知られているテキストコーパス中の誤りを検出する試みとしては、形態素コーパス中での過分割の誤りを検出する方法（内山将夫、「形態素解析結果から過分割を検出する統計的尺度」、言語処理学会会誌、Ｖｏｌ．６、Ｎｏ．７、（１９９９））などがある。しかし、従来の手法の多くは、誤りの種類を特化し、汎用性の見えにくい手法である。
【０００４】そこで、本件出願人らによって、一般的にどの問題に対しても用いることができると考えられている用例ベース手法や、決定リスト手法を利用した、対象とするコーパスのみから間違っている確率を算出し、誤りを検出する手法が考え出された。（村田真樹、内山将夫、内元清貴、馬青、井佐原均、「決定リスト、用例ベース手法を用いたコーパス誤り検出・誤り訂正」、情報処理学会　自然言語処理研究会、２０００－ＮＬ－１３６、ｐｐ．４９－５６（２０００））しかし、これら従来の手法でも、学習の前に誤りの検出を行わなければならず、いわばオフラインでの検出手法である上に、検出処理も、誤りがありそうな部分への絞り込み作業なしに、全コーパスを対象に１語ずつ調べていくため、好適な検出効率を得ることは難しかった。このため、大規模なテキストコーパスの場合には、検出が難しく、費用コスト的にも大きくなる問題があった。
【０００５】
【発明が解決しようとする課題】本発明は、上記従来技術の有する問題点に鑑みて創出されたものであり、その目的は、高速かつ高効率でテキストコーパス中の誤りを検出する方法を創出することである。ひいては、言語処理技術の向上に寄与することを目的とする。
【０００６】
【課題を解決するための手段】本発明は、上記の課題を解決するために、次のようなテキストコーパスの誤り検出方法を創出する。すなわち、単語情報を含む予め作成されたテキストコーパスにおける該単語情報の誤りを検出する方法であって、該各単語情報の分類をニューラルネットワークにおけるクラスとして捉え、それらを小規模な２クラス問題に分割して、複数のモジュールを構成する。
【０００７】そして、各モジュールがニューラルネットワークにおける学習過程において収束するか否かの演算を行い、収束しない場合に、該モジュールに該単語情報の誤りがあると判定し、該モジュールを抽出する。このように本発明では、一般にニューラルネットワークでは問題とされていた収束しない現象を逆手にとって、収束しないときに誤りであると判定する誤り検出方法を創出した。
【０００８】ここで、前記単語情報が、品詞に係る情報

であって、該情報をタグ形式でテキスト中に埋め込み、テキストコーパスを構成し、該タグの誤りを検出する場合にも適用することができる。

【０００９】又、本発明は以上の処理を行うことによって、テキストコーパスの誤りを検出可能な装置として提供することもできる。この時、誤りを検出することに特化した機能を有する装置として提供してもよいし、他の装置の一部として構成することもできる。例えば、テキストコーパスを手動、半自動、自動的に作成する装置に組み込んで、作成後のコーパスの誤りを連続的に検出することもできるし、その他、任意の言語処理を行う装置に付加することもできる。

【００１０】

【発明の実施の形態】以下、本発明の実施方法を図面に示した実施例に基づいて説明する。なお、本発明の実施形態は以下に限定されず、適宜変更可能である。以下においては、テキストコーパスの一例として、日本語によるコーパスを挙げて説述していくが、本発明の実施方法は、性質上実現出来ない場合を除き、英語、中国語、韓国語等のいかなる言語に対しても適用可能である。また、本発明が対象とするテキストコーパスは、品詞や形態素区切り等、任意の単語情報を含むテキストコーパスであってよく、本発明はそれらの単語情報に係る誤りを効果的に検出できる方法である。

【００１１】種々の学習システムはそれぞれの用途にあわせ、大規模データベースを学習することによって必要な知識を獲得する。大規模データベースは通常、人手で作成されるため、細心な注意を払ってもエラーは存在する。データベースの品質を高めるため、そして、それを用いる学習するシステムの性能を高めるため、エラーを自動検出する技術が必要である。しかし、これまで開発されたエラー自動検出はすべてそのデータを利用する前にあらかじめ（つまり、オフライン的に）行うものであり、常に全データを対象にデータを一つずつチェックして行わなければならない。そのため、例えばテキストコーパスのような大規模データベースの場合、その計算コストは非常に高い。

【００１２】そこで、本発明において、テキストコーパスの誤りを検出する際、次の特徴を有する誤り検出方法及び装置を提供する。

（１）　エラーは、そのデータを学習する最中に検出される。

（２）　検出は全データをスキャンするのではなく、エラーのあるごく小さなデータエリアに直接飛びつき、それらのエリアに絞って行う。

（３）　データが修正された後の再学習は、全データを対象に行う必要がなく、修正がかかったエリアのデータのみに対し行えばよい。

（４）　学習機械としては、大規模で複雑な学習問題(データ)を多数の小規模で簡単な問題(データセット)に分割して学習が行えるモジュール型ニューラルネットが用いられる。

（５）　上記ニューラルネットを構成する個々のモジュールは、学習データにエラーがなければ必ず収束する、という前提に基づき処理を行う。

（６）　上記（１）ないし（３）を可能にしたのは上記（５）の特性を逆に利用した結果である。つまり、収束しないモジュールにエラーデータが存在すると考え、エラー検出を行う。

【００１３】図１には上記の本発明の特徴を、テキストコーパスの誤り検出に用いた場合の処理概念図を示す。まず、大規模テキストコーパス（１０）を、学習対象となる学習データセット（１１）として用いる。該学習データセットは大規模で複雑な学習問題であるから、これを多数の小規模で簡単なデータサブセット（１２ａ）（１２ｂ）（１２ｃ）（１２ｄ）（１２ｅ）・・に分割する。この分割によってモジュール型ニューラルネットが形成される。

【００１４】そして、各データサブセット（１２ａ・・・）をそれぞれ学習に用いられ、その際、収束（１３）すれば当該データサブセットに誤りはない（１４）と判断される。一方、例えばあるデータサブセット（１２ｃ）において学習が収束しなかった（１３’）場合、データサブセット（１２ｃ）を抽出（１５）し、例えば人手によって修正（１６）を施す。修正方法は自動的な手法によってもよいし、任意である。

【００１５】修正（１６）を施されたデータサブセット（１２ｃ）は再び、再学習（１７）に用いられ、学習が収束（１８）し、データに誤りはない（１９）として処理される。本発明は、以上の流れにしたがってテキストコーパスの誤り検出を行うものであり、以下に、これらの本発明が特徴とする革新的アイデアに基づいて創出された誤り検出方法を詳細に説明する。

【００１６】人手で作成されたコーパスには、単純ミス型（例えば，「動詞」を「同士」と表記してしまう場合など。）、不正確な知識による知識型（例えば，「国立」という地名もあるのに、「国立」をすべて普通名詞にしてしまう場合など。）、矛盾型（例えば、格助詞であるべき「の」をときには接続助詞にしてしまう場合など。）という三種類の誤りが考えられる。

【００１７】単純ミス型誤りは、電子辞書や品詞体系リストなどを参照すれば容易に検出可能である。一方、知識型誤りの検出は自動的な方法では困難である。品詞タグ付けを一種の入出力マッピング問題として捉えるなら、矛盾型誤りを同じ入力を持ちながら出力が異なるデータの集合として考えることができる。このような誤りを検出する手法は従来から幾つか提案されている。しかし、それらの手法はいずれも学習の前に行わなければならないものであり、いわばオフラインで行う手法であった。そして、検出処理は、誤りがありそうな部分への絞

り作業なしに、全コーパスを対象に一語ずつ調べて行くため、計算処理に時間と費用がかかり、非効率的である。とくに、コーパスが高精度である場合には、全コーパスを検査する手法は、非常に無駄な作業が多い。

【００１８】そこで、本発明においては、次のような誤り検出方法を創出した。まず、学習問題とされる品詞タグ付けは、任意の文

$W = w_1 w_2 \cdots w_s$

が与えられたとき、マッピング処理或いはクラス分け問題

$W_p \rightarrow \tau_p$

によって品詞列

$T = \tau_1 \tau_2 \cdots \tau_s$

を見つけることである。ただし、ｐは品詞を定めようとする目標言語のコーパスにおける位置を表し、W_pは目標単語w_pを中心とした左右それぞれ（ｌ，ｒ）個の単語で構成される単語列である。すなわち、

$W_p = w_{p-l} \cdots w_p \cdots w_{p+r}$

となる。ただし、$p - l \geqq s_s$、$p + r \leqq s_s + s$、s_sは文頭単語の位置である。

【００１９】誤りの検出の一例として、例えば京大テキストコーパスからすくなくとも一箇所の誤りを持つ２１７文を用いる。それらの文はのべ６８１６個の単語（うち、異なりが２４１０ 個）を持ち、９７種類の品詞を有する。品詞をクラスとして捉えるなら，この品詞タグ付け問題は９７classのクラス分け問題となる。

【００２０】この９７class問題をまず一意的に４６５６個のtwo-class問題に分割する。そして、まだ学習データの多いtwo-class問題はその数が８０以下になるように更に無作為に分割する。このようにして、この97class問題が23,231個の小規模で簡単なtwo-class問題に分割される。

【００２１】本発明で用いるM^3ネットワークでは、これらの問題はそれぞれ独立のモジュールで学習される。そして、学習したモジュールはMINやMAXなどの簡単な演算で統合され、品詞タグ付けが行われる。各モジュールへの入力ベクトルＸは単語列W_pから以下のように構成される。

【式１】$X = (x_{p-l}, \cdots, x_p, \cdots, x_{p+r})$

要素x_pは目標単語を符号化するω次元のbinary-codedベクトルである。一方、文脈にある単語に対応する要素x_t（$t \neq p$）はその単語に付与された品詞を符号化するτ次元のbinary-codedベクトルである。目標出力Ｙは目標単語に付与すべき品詞を符号化するτ次元の binary-codedベクトルであり、式２によって示される。

【式２】$Y = (y_1, y_2, \cdots, y_r)$

【００２２】M^3ネットワークは簡単で小規模な問題を扱う多数のモジュールから構成されるため、収束性の問題が基本的に生じない。言い替えれば、あるモジュールが学習において収束しなければ、このモジュールの学習データに矛盾型誤りがあると考えられる。即ち、学習データセットに、式３の条件を満足するデータのペア(X_i, Y_i)と(X_j, Y_j)が存在する。

【式３】$X_i = X_j$，$Y_i \neq Y_j$　（$i \neq j$）

ただし、X_iとX_jは入力であり、Y_iとY_jはそれぞれ対応する目標出力である。従って、このタイプの誤りは学習の時に、収束しないモジュールだけを選び出すことによって検出できる。

【００２３】本発明によるこの検出方法は全データをスキャンするのではなく、誤りのあるごく小さなデータエリアにスキップし、それらのエリアに絞って行う、と見ることができる。このような検出法は、従来のように全コーパスを調べていくよりもはるかに効率的であって、特に高精度のコーパスを用いている場合には、誤りデータが極少数のモジュールに限られているため、非常に有効である。

【００２４】そして、誤りが訂正された後の再学習は、全コーパスを対象に行う必要がなく、また全モジュールに対し行う必要もなく、修正がかけられたエリアのデータのみを用い、収束しないモジュールだけに対して行えばよい。いうまでもなく、このような低コストの再学習は学習システムの性能向上にも寄与することができる。

【００２５】本発明では、以上の手法を用いたテキストコーパスの誤り検出装置を提供することができる。該手法は、高速な処理を行うことが可能なため、テキストコーパスによる言語処理装置に本発明の装置を付加し、学習を行うと同時に、誤りを検出し、より高精度な言語処理装置を提供することも可能となる。また、テキストコーパスを人手によって作成する装置、手動及び自動的に作成する装置、言語固有情報を自動獲得して、自動的に作成する装置等に、本発明の装置を付加し、テキストコーパスの作成と同時に誤りを検出し、高精度なテキストコーパスを完成させることもできる。

【００２６】以下、本発明による検出方法の検出精度を示す。単語と品詞を表すベクトルの次元ω，τはそれぞれ１６と８に設定した。単語列の長さ（ｌ，ｒ）は（２，２）に設定した。従って、モジュールの入力層のユニット数は

$[(l+r) \times \tau] + [1 \times \omega] = 48$

であった。

【００２７】初期値としてすべてのモジュールは48-2-1の３層パーセプトロンで構成された。一回の学習は目標誤差0.05に達した時点あるいは5000ステップまで行われる。目標誤差まで到達できないモジュールについては、中間層ユニット数を２個ずつ増やし、再度学習を最大５回まで行った。

【００２８】図２は実験結果を示している。23231個のモジュールのうち、僅か８２個が収束しなかった。このうち、８１個の中に矛盾なデータがあった。矛盾なデータは計９７ペアであった。この９７ペアの中、９４ペア

にそれぞれ１個ずつの誤りデータがあった。図３は正しく検出された例を示す。左列の数字はそれぞれ検出された単語が所在する文番号と文の中の位置を示す。下線のある単語はチェックを受けている単語で、×が付いているほうがその単語に付与されている品詞が誤っていることを示す。

【００２９】一方、正しく検出されなかった残りの３ペアはすべて助詞／判定詞の「で」についてのものであった。しかし実際の文を調べた結果、これらの「で」を判定するためには文全体、すなわち構文情報を用いる必要があることが分かった。従って、本実験結果は、本発明による検出方法の精度は、構文情報を必要としない誤りの検出において実質的に１００％に達したことを意味している。

【００３０】このように、ニューラルネットワークにおける収束しない現象を用いたテキストコーパスの誤り検出方法は、極めて高精度、高信頼性を有し、しかも従来の手法に比して飛躍的に高速な方法を実現している。上記実験では小規模なテキストコーパスであったが、より大規模な場合には、さらにその効果が顕著になると考えられ、本発明の有用性が証明された。

【００３１】

【発明の効果】本発明は、以上の構成を備えるので、次の効果を奏する。請求項１に記載のテキストコーパスの誤り検出方法によると、ニューラルネットを用いるときによく悩まされる収束しない問題を逆手に取り、人手で作成したコーパスを学習しながらその中に含まれる誤りを収束しないモジュールを調べることによって高効率に検出する手法を実現することができる。これによって、高速かつ高精度、低コストな検出方法に寄与する。

【００３２】請求項２に記載のテキストコーパスの誤り検出方法によると、テキストコーパスに広く用いられているタグを利用したテキストコーパスに本発明の方法を用いることができるので、該手法を有効に活用することができる。

【００３３】請求項３に記載のテキストコーパスの誤り検出装置によると、オンラインで高効率に誤りを検出することで、正確なテキストコーパスでの学習に寄与し、さらに誤りの訂正、訂正後のコーパスによる再学習が効率よく行える。これによって、学習システムの性能向上を図ることができ、ひいては言語処理技術の向上に寄与する。

【００３４】請求項４に記載のテキストコーパスの誤り検出装置によると、広く普及したタグ形式のテキストコーパスを用いることができるので、有用性が高い。

【図面の簡単な説明】

【図１】本発明の処理概要図である。

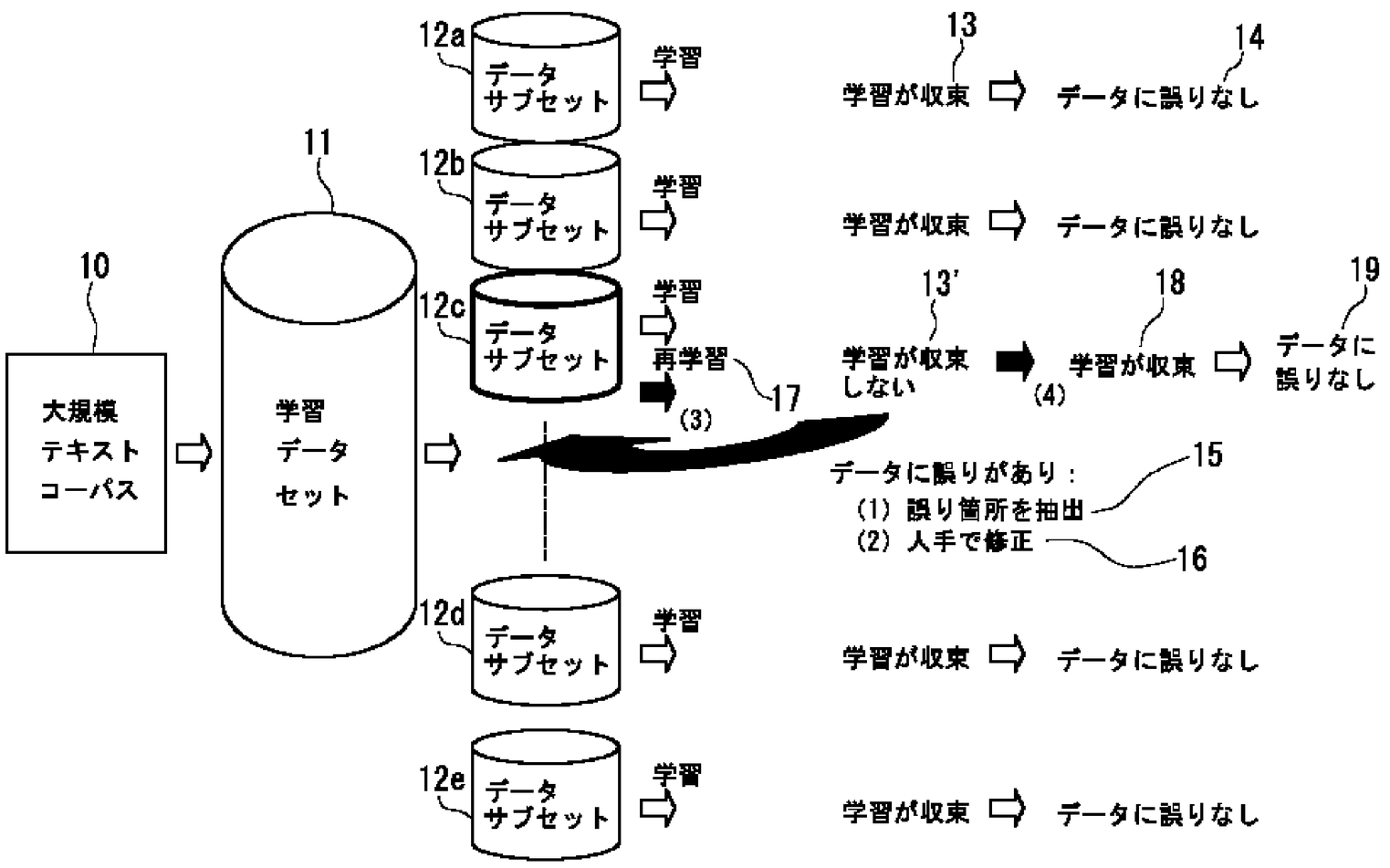
【図２】誤り検出の実験結果である。

【図３】誤りの検出例である。

【符号の説明】

１０	大規模テキストコーパス
１１	学習データセット
１２ａないしｅ	データサブセット
１３	学習の収束結果
１３’	学習の非収束結果
１４	データに誤りなし
１５	データに誤りがあった場合のサブセット抽出過程
１６	修正過程
１７	再学習過程
１８	再学習後の収束結果
１９	データに誤りなし

【図１】



【図２】

モジュールの総数	収束しないモジュールの数	矛盾データを有すモジュールの数	矛盾データの数	誤りの数
23,231	82	81	97	94

【図３】

2-18	× **政治家**:名詞:普通名詞:*	**側**:接尾辞:名詞性名詞接尾辞: *	**の**:助詞:格助詞:*	**おだて**:名詞:普通名詞:*	**に**:助詞:格助詞:*
124-19	○**シナリオ**:名詞:普通名詞:*	**づくり**:接尾辞:名詞性名詞接尾辞:*	**の**: 助詞:接続助詞:*	**方向**:名詞:普通名詞:*	**を**:助詞:格助詞:*