(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2016-115288
(P2016-115288A)
(43)公開日　平成28年6月23日(2016.6.23)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 17/30	(2006.01)	G06F	17/30	414Z	
		G06F	17/30	170A	

審査請求 未請求　請求項の数 12　ＯＬ　（全 24 頁）

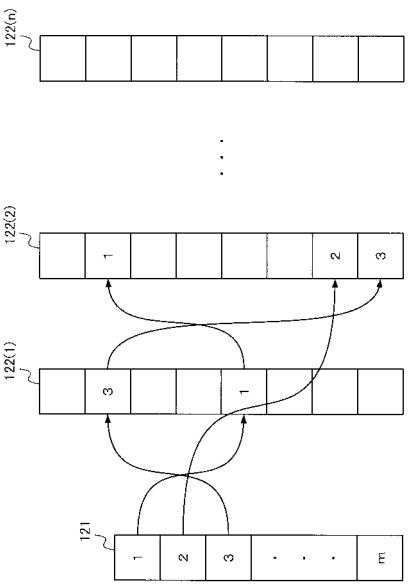
(21)出願番号　特願2014-255623 (P2014-255623)
(22)出願日　平成26年12月17日 (2014.12.17)

(71)出願人　390009531
インターナショナル・ビジネス・マシーンズ・コーポレーション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国１０５０４　ニューヨーク州　アーモンク　ニュー　オーチャード　ロード
New Orchard Road, Armonk, New York 10504, United States of America
(74)代理人　100108501
弁理士　上野　剛史

最終頁に続く

(54)【発明の名称】データを集計するシステム、方法およびプログラム

(57)【要約】
【課題】集計処理において、必要なメモリ領域（記憶容量）を削減し、かつ、対象データや属性の更新処理に要する手間を削減する。
【解決手段】データを集計するシステムであって、対象データごとに所定の順序で並び、対象データの識別情報から対象データが有する属性のリストを指定するＤｔｏＫインデックス１２２と、複数の対象データが有する属性のリストである単語リスト１２１と、を記憶するインデックス記憶部と、属性ごとにこの属性を有する対象データを調べ、対象データとの関係が所定の基準を満たす属性を集計する集計処理を行う集計処理部と、を備える。そして、単語リスト１２１の各属性に関して、対象データごとの第１リストの要素を順に辿るリンクが設定されており、集計処理部は、属性ごとに設定されたリンクに基づき、属性を有する対象データを調べる。
【選択図】図２



【特許請求の範囲】
【請求項１】
データを集計するシステムであって、
集計処理の対象である対象データの識別情報から当該対象データが有する属性のリストである第１リストを指定するインデックスであって、当該対象データごとの当該第１リストは所定の順序で並び、当該第１リストの各要素が当該属性の情報を含む、当該インデックスと、複数の当該対象データが有する属性のリストである第２リストと、を記憶するインデックス記憶部と、
前記属性ごとに当該属性を有する前記対象データを調べ、当該対象データとの関係が所定の基準を満たす属性を集計する集計処理を行う集計処理部と、を備え、
前記第２リストの各属性に関して、前記対象データごとの前記第１リストの要素を順に辿るリンクが設定されており、
前記集計処理部は、前記属性ごとに設定された前記リンクに基づき、当該属性を有する前記対象データを調べる、システム。
【請求項２】
前記リンクは、前記対象データごとの前記第１リストの各要素が、当該要素に含まれるのと同一の前記属性の情報を含む最も近い後続の第１リストの要素を特定する情報を含むことにより、設定される、請求項１に記載のシステム。
【請求項３】
前記第２リストの各属性は、複数の前記対象データにおける個々の対象データが当該属性を有する頻度の降順にソートされている、請求項１に記載のシステム。
【請求項４】
前記対象データごとの前記第１リストの各要素は、当該要素が含む情報により特定される前記属性に対して付与された値をさらに含み、
前記第１リストの各要素は、前記値に基づく順序でソートされている、請求項１に記載のシステム。
【請求項５】
前記属性に対して付与された値は、当該属性と当該属性を有する前記対象データとの関係に基づく重み値であり、
前記第１リストの各要素は、前記値の降順にソートされている、請求項４に記載のシステム。
【請求項６】
前記対象データごとの前記第１リストは、最も古い第１リストを最後尾として作成された順に並び、新規に作成された第１リストを追加する場合は、第１リストの並びの先頭に配置され、かつ、新規に作成された当該第１リストの各要素に含まれる属性に関するリンクが更新される、請求項１に記載のシステム。
【請求項７】
前記対象データごとの前記第１リストは、最も古い第１リストを最後尾として作成された順に並び、最も古い第１リストを削除する場合は、最後尾の第１リストが削除され、かつ、削除された当該第１リストの各要素に含まれる属性に関するリンクが更新される、請求項１に記載のシステム。
【請求項８】
データを集計するシステムであって、
集計処理の対象である対象データごとに当該対象データが有する属性を登録したリストを含むインデックスを記憶するインデックス記憶部と、
前記属性ごとに当該属性を有する前記対象データを調べ、当該対象データとの関係が所定の基準を満たす属性を集計する第１手法、および、前記対象データごとに当該対象データが有する前記属性を調べ、当該対象データとの関係が所定の基準を満たす属性を集計する第２手法のどちらで集計処理を行うかを判定する判定部と、
前記判定部による判定に応じて前記第１手法および前記第２手法の一方により集計処理

を行う集計処理部と、を備え、
前記インデックスは、相異なる前記対象データが有する同一の前記属性どうしを関連付けており、
前記集計処理部は、前記第１手法により集計処理を行う場合に、前記インデックスにおける同一の前記属性どうしの関連に基づき、当該属性を有する前記対象データを調べる、システム。
【請求項９】
データを集計する方法であって、
集計処理の対象である対象データの識別情報から当該対象データが有する属性のリストである第１リストを指定するインデックスであって、当該対象データごとの当該第１リストは所定の順序で並び、当該第１リストの各要素が当該属性の情報を含む、当該インデックスと、複数の当該対象データが有する属性のリストである第２リストと、前記第２リストの各属性に関して、前記対象データごとの前記第１リストの要素を順に辿るリンク構造と、を記憶するインデックス記憶部と、
前記属性の集計処理を行う集計処理部と、を備えるシステムにおいて、
前記集計処理部が、
前記属性ごとに設定された前記リンクに基づき、当該属性を有する前記対象データの数を調べるステップと、
前記属性を、調べた対象データの数が多い上位の所定数の属性を集計し、集計結果を出力するステップと、
を含む、方法。
【請求項１０】
データを集計する方法であって、
集計処理の対象である対象データごとに当該対象データが有する属性を登録したリストを含むインデックスを記憶するインデックス記憶部と、
前記属性の集計処理を行う集計処理部と、
前記集計処理部により行われる集計処理の手法を判定する判定部と、を備えるシステムにおいて、
前記判定部が、前記属性ごとに当該属性を有する前記対象データを調べ、当該対象データとの関係が所定の基準を満たす属性を集計する第１手法、および、前記対象データごとに当該対象データが有する前記属性を調べ、当該対象データとの関係が所定の基準を満たす属性を集計する第２手法のどちらで集計処理を行うかを判定するステップと、
前記判定部が前記第１手法により集計処理を行うと判定した場合に、前記集計処理部が、相異なる前記対象データが有する同一の前記属性どうしを関連付けてなる前記インデックスにおける同一の前記属性どうしの関連に基づき、当該属性を有する前記対象データを調べ、当該対象データとの関係が所定の基準を満たす属性を集計するステップと、
前記判定部が前記第２手法により集計処理を行うと判定した場合に、前記集計処理部が、前記インデックスに基づき、各対象データが有する前記属性を調べ、当該対象データとの関係が所定の基準を満たす属性を集計するステップと、
を含む、方法。
【請求項１１】
コンピュータを、
集計処理の対象である対象データの識別情報から当該対象データが有する属性のリストである第１リストを指定するインデックスであって、当該対象データごとの当該第１リストは所定の順序で並び、当該第１リストの各要素が当該属性の情報を含む、当該インデックスと、複数の当該対象データが有する属性のリストである第２リストと、を記憶するインデックス記憶手段と、
前記属性ごとに当該属性を有する前記対象データを調べ、当該対象データとの関係が所定の基準を満たす属性を集計する集計処理を行う集計処理手段として機能させ、
前記第２リストの各属性に関して、前記対象データごとの前記第１リストの要素を順に

辿るリンクが設定されており、
前記集計処理手段の機能として、前記属性ごとに設定された前記リンクに基づき、当該属性を有する前記対象データを調べる、プログラム。
【請求項１２】
コンピュータを、
集計処理の対象である対象データごとに当該対象データが有する属性を登録したリストを含み、相異なる前記対象データが有する同一の前記属性どうしを関連付けてなるインデックスを記憶するインデックス記憶手段と、
前記属性ごとに当該属性を有する前記対象データを調べ、当該対象データとの関係が所定の基準を満たす属性を集計する第１手法、および、前記対象データごとに当該対象データが有する前記属性を調べ、当該対象データとの関係が所定の基準を満たす属性を集計する第２手法のどちらで集計処理を行うかを判定する判定手段と、
前記判定手段による判定に応じて前記第１手法および前記第２手法の一方により集計処理を行う集計処理手段として機能させ、
前記集計処理手段の機能として、前記第１手法により集計処理を行う場合に、前記インデックスにおける同一の前記属性どうしの関連に基づき、当該属性を有する前記対象データを調べる、プログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、データを集計するシステム、方法およびプログラムに関し、特に所定の属性を有する対象データの集合中から所定の属性を集計する技術に関する。
【背景技術】
【０００２】
複数種類の属性を持つ大量のデータに対する集計操作の一つに、処理対象として指定されたデータ（以下、対象データ）の集合中に出現する属性を、ある規則に基づいて各属性に付与された値である点数（重み、スコア）の高いものから順にｋ種類求める集計操作がある。この集計操作は、Ｔｏｐ－ｋ等とも呼ばれる。例えば、対象データが文書（テキスト）で、属性がその文書に含まれる単語である場合を考える。また、属性である各単語に付与される点数を全文書中の出現数とする。このとき、特定の条件を満たす複数の文書だけに出現する単語のうち、出現頻度上位ｋ個の単語を求める操作は、Ｔｏｐ－ｋの集計操作の一例である。
【０００３】
この種の従来技術として、テキストの識別情報から当該テキストに含まれるキーワードのリストを指定する第１インデックス（ＤＯＣ＿ＴＯ＿ＫＥＹインデックス）と、キーワードの識別情報から当該キーワードを含むテキストのリストを指定する第２インデックス（ＫＥＹ＿ＴＯ＿ＤＯＣインデックス）とを用いて、テキストを検索する技術がある（特許文献１参照）。ここで、テキストは上記の「対象データ」の一例であり、キーワードは上記の「属性」の一例である。この従来技術では、検索条件の入力を受け付けると、まず、第１インデックスによる検索時間および第２インデックスによる検索時間を見積もる。そして、より高速であると判断されたインデックスを用いて検索する。
【先行技術文献】
【特許文献】
【０００４】
【特許文献１】特開２００７－１５６７３９号公報
【発明の概要】
【発明が解決しようとする課題】
【０００５】
ＫＥＹ＿ＴＯ＿ＤＯＣインデックス（以下、ＫｔｏＤインデックスと略記する）を用いた処理は、例えば、次のような手順による処理となる。すなわち、まず、出現頻度の高い

順に属性を順次選択する。そして、選択された属性を含む対象データのリストが、検索条件を満たすか否かを判断する。そして、検索条件を満たす対象データの数が多い順にｋ個の属性を選択する。これにより、選択された属性が集計結果となる。しかし、この処理は、集計される対象となる属性の種類が多い場合、多大な処理時間を要する。

【０００６】

ＤＯＣ＿ＴＯ＿ＫＥＹインデックス（以下、ＤｔｏＫインデックスと略記する）を用いた処理は、例えば、次のような手順による処理となる。すなわち、まず、検索条件を満たす対象データを選択する。そして、選択された対象データの識別情報に対応する属性のリストを取得する。そして、リストに示された属性を含む対象データの数を属性毎に集計する。しかし、この処理は、対象データの数が多い場合、多大な処理時間を要する。

【０００７】

特許文献１の従来技術は、上記２種類のインデックスを有し、各インデックスを用いた処理に要する時間を見積もり、より短時間で実行可能と判断した処理を実行するものである。しかし、この従来技術は、ＫｔｏＤインデックスとＤｔｏＫインデックスの両方を有するため、インデックスを保持するために必要なメモリ領域（記憶容量）が大きい。

【０００８】

また、ＫｔｏＤインデックスに関しては、対象データが追加された場合、追加された対象データおよびその属性に対応させるためにはインデックスを作成しなおさなければならず、更新処理が煩雑であった。

【０００９】

そこで本発明は、上記の集計処理を行う技術において、ＫｔｏＤインデックスとＤｔｏＫインデックスの両方を用意する場合と比較して、必要なメモリ領域（記憶容量）を削減し、かつ、対象データや属性の更新処理に要する手間を削減することを目的とする。

【課題を解決するための手段】

【００１０】

上記の目的を達成するため、本発明は、次のようなシステムとして実現される。このシステムは、データを集計するシステムであって、集計処理の対象である対象データの識別情報から対象データが有する属性のリストである第１リストを指定するインデックスであって、対象データごとの第１リストは所定の順序で並び、第１リストの各要素が属性の情報を含む、インデックスと、複数の対象データが有する属性のリストである第２リストと、を記憶するインデックス記憶部と、属性ごとにこの属性を有する対象データを調べ、対象データとの関係が所定の基準を満たす属性を集計する集計処理を行う集計処理部と、を備える。そして、第２リストの各属性に関して、対象データごとの第１リストの要素を順に辿るリンクが設定されており、集計処理部は、属性ごとに設定されたリンクに基づき、属性を有する対象データを調べる。

【００１１】

より好ましくは、第１リストにおけるリンクは、対象データごとの第１リストの各要素が、この要素に含まれるのと同一の属性の情報を含む最も近い後続の第１リストの要素を特定する情報を含むことにより、設定される。

【００１２】

さらに好ましくは、第２リストの各属性は、複数の対象データにおける個々の対象データが属性を有する頻度の降順にソートされている。

【００１３】

さらに好ましくは、対象データごとの第１リストの各要素は、この要素が含む情報により特定される属性に対して付与された値をさらに含む。そして、第１リストの各要素は、この値に基づく順序でソートされている。

【００１４】

より詳細には、属性に対して付与された値は、この属性とこの属性を有する対象データとの関係に基づく重み値であり、第１リストの各要素は、この値の降順にソートされている。

【００１５】
さらに好ましくは、対象データごとの第１リストは、最も古い第１リストを最後尾として作成された順に並び、新規に作成された第１リストを追加する場合は、第１リストの並びの先頭に配置され、かつ、新規に作成された第１リストの各要素に含まれる属性に関するリンクが更新される。
【００１６】
さらに好ましくは、対象データごとの第１リストは、最も古い第１リストを最後尾として作成された順に並び、最も古い第１リストを削除する場合は、最後尾の第１リストが削除され、かつ、削除された第１リストの各要素に含まれる属性に関するリンクが更新される。
【００１７】
また、本発明は、次のようなシステムとしても実現される。このシステムは、データを集計するシステムであって、集計処理の対象である対象データごとに対象データが有する属性を登録したリストを含むインデックスを記憶するインデックス記憶部と、属性ごとにこの属性を有する対象データを調べ、対象データとの関係が所定の基準を満たす属性を集計する第１手法、および、対象データごとに対象データが有する属性を調べ、対象データとの関係が所定の基準を満たす属性を集計する第２手法のどちらで集計処理を行うかを判定する判定部と、判定部による判定に応じて第１手法および第２手法の一方により集計処理を行う集計処理部と、を備える。そして、インデックスは、相異なる対象データが有する同一の属性どうしを関連付けており、集計処理部は、第１手法により集計処理を行う場合に、インデックスにおける同一の属性どうしの関連に基づき、この属性を有する対象データを調べる。
【００１８】
また、本発明は、次のような方法としても実現される。この方法は、データを集計する方法であって、集計処理の対象である対象データの識別情報から対象データが有する属性のリストである第１リストを指定するインデックスであって、対象データごとの第１リストは所定の順序で並び、第１リストの各要素がこの属性の情報を含む、インデックスと、複数の対象データが有する属性のリストである第２リストと、第２リストの各属性に関して、対象データごとの第１リストの要素を順に辿るリンク構造と、を記憶するインデックス記憶部と、属性の集計処理を行う集計処理部と、を備えるシステムにおいて、集計処理部が、属性ごとに設定されたリンクに基づき、この属性を有する対象データの数を調べるステップと、属性を、調べた対象データの数が多い上位の所定数の属性を集計し、集計結果を出力するステップと、を含む。
【００１９】
また、本発明は、次のような方法としても実現される。この方法は、データを集計する方法であって、集計処理の対象である対象データごとにこの対象データが有する属性を登録したリストを含むインデックスを記憶するインデックス記憶部と、属性の集計処理を行う集計処理部と、この集計処理部により行われる集計処理の手法を判定する判定部と、を備えるシステムにおいて、判定部が、属性ごとにこの属性を有する対象データを調べ、対象データとの関係が所定の基準を満たす属性を集計する第１手法、および、対象データごとにこの対象データが有する属性を調べ、対象データとの関係が所定の基準を満たす属性を集計する第２手法のどちらで集計処理を行うかを判定するステップと、判定部が第１手法により集計処理を行うと判定した場合に、集計処理部が、相異なる対象データが有する同一の属性どうしを関連付けてなるインデックスにおける同一の属性どうしの関連に基づき、この属性を有する対象データを調べ、対象データとの関係が所定の基準を満たす属性を集計するステップと、判定部が第２手法により集計処理を行うと判定した場合に、集計処理部が、インデックスに基づき、各対象データが有する属性を調べ、対象データとの関係が所定の基準を満たす属性を集計するステップと、を含む。
【００２０】
さらにまた、本発明は、コンピュータを制御して上述した装置の各機能を実現するプロ

グラム、あるいは、コンピュータに上記の各ステップに対応する処理を実行させるプログラムとしても実現される。このプログラムは、磁気ディスクや光ディスク、半導体メモリ、その他の記録媒体に格納して配布したり、ネットワークを介して配信したりすることにより、提供することができる。

【発明の効果】

【００２１】

本発明によれば、ＫｔｏＤインデックスを別途作成することなく、集計対象の属性ごとに、その属性を有する対象データの数を調べ、対象データ数の多い上位の所定数の属性を集計する。これにより、ＫｔｏＤインデックスとＤｔｏＫインデックスの両方を用意する場合と比較して、必要なメモリ領域（記憶容量）を削減し、かつ、対象データや属性の更新処理に要する手間を削減することが可能となる。

【図面の簡単な説明】

【００２２】

【図１】本実施形態による集計システムの構成を示す図である。

【図２】本実施形態のインデックス記憶部に記憶されるインデックスの構成を示す図である。

【図３】第１の手法により集計処理を行う場合の集計処理部の動作を示すフローチャートである。

【図４】第１の手法により集計処理を行う場合の集計処理部の動作を示すフローチャートである。

【図５】ＤｔｏＫインデックスの作成処理の手順を示すフローチャートである。

【図６】新規文書追加処理の手順を示すフローチャートである。

【図７】Ｊａｖａプログラミング言語を用いたＤｔｏＫインデックスの実装例を示す図であり、図７（ａ）はＤｔｏＫインデックスのエントリを設定するコード、図７（ｂ）はＤｔｏＫリストを生成するコード、図７（ｃ）は新しいＤｔｏＫインデックスを追加するコード、図７（ｄ）はＤｔｏＫインデックスを削除するコードを示す図である。

【図８】Ｊａｖａプログラミング言語を用いたＤｔｏＫインデックスの実装例を示す図であり、図７に示すコードにより設定されたリンクの様子を示す図である。

【図９】Ｊａｖａプログラミング言語を用いたＤｔｏＫインデックスの他の実装例を示す図であり、ＤｔｏＫインデックスのエントリを設定するコードを示す図である。

【図１０】Ｊａｖａプログラミング言語を用いたＤｔｏＫインデックスの他の実装例を示す図であり、図９に示すコードにより設定されたリンクの様子を示す図である。

【図１１】図９～図１０に示す実装例によるインデックスの構成を示す図である。

【図１２】図９～図１１の実装例において、ＤｔｏＫインデックスを削除する処理の手順を示すフローチャートである。

【図１３】図１１に示したインデックス構成において、ＤｔｏＫインデックスを削除した様子を示す図である。

【図１４】本実施形態の集計システムを構成するのに好適なハードウェア構成例を示す図である。

【発明を実施するための形態】

【００２３】

以下、添付図面を参照して、本発明の実施形態について詳細に説明する。本実施形態による集計システムは、複数種類の属性を持つ対象データの集合中に出現する属性を、何らかの規則に基づいて所定の基準個数だけ求める集計処理を行う。以下では、対象データの一例として、検索等の何らかの手段で処理対象として絞り込まれた文書、属性の一例として、各文書に含まれる単語を用いた場合を例として説明する。また、この例において、集計の規則は、絞り込まれた文書全体における出現数の多いものから順に基準個数（ｋ個）の単語を選択するものとする。

【００２４】

＜システム構成＞

図１は、本実施形態による集計システムの構成を示す図である。図１に示すように、本実施形態の集計システム１００は、文書ＤＢ（データベース）２００に接続されている。文書ＤＢ２００は、処理対象となり得る文書を格納している。集計処理の対象となる文書は、例えば、文書ＤＢ２００に格納されている文書のうちで、何らかの検索条件に基づく検索等により指定された（絞り込まれた）文書とすることができる。

【００２５】

本実施形態では、集計システム１００は、文書ＤＢ２００に格納されている文書に関して、何らかの検索条件に基づく検索により集計処理の対象となる文書を指定する（絞り込む）。例えば、特定の単語を含む文書を検索し、得られた文書を集計処理の対象とする。そして、得られた文書に含まれる単語のうち、出現頻度の高いものから順に、予め定められた個数（ｋ個）の単語を求める。図１に示すように、本実施形態の集計システム１００は、入力受け付け部１１０と、インデックス記憶部１２０と、集計処理部１３０と、判定部１４０と、インデックス更新部１５０とを備える。

【００２６】

入力受け付け部１１０は、集計処理を行うための入力を受け付ける。具体的には、入力受け付け部１１０は、集計処理の対象となる文書を指定するための絞り込み条件（検索条件等）の入力を受け付ける。また、入力受け付け部１１０は、集計しようとする属性を指定する情報の入力を受け付ける。

【００２７】

インデックス記憶部１２０は、集計処理に用いられるインデックスを記憶している。本実施形態では、文書の識別情報からその文書に含まれる単語のリストを指定するＤｔｏＫインデックスが用いられる。ＤｔｏＫインデックスは、文書ＤＢ２００に格納されている文書ごとに作成されている。また、本実施形態では、ＤｔｏＫインデックスにおける文書ごとの単語のリストにおいて、各文書に出現する同一の単語間にリンクを設定している。言い換えれば、相異なる文書のＤｔｏＫインデックスに含まれる同一の単語どうしを関連付けている。これにより、例えば、ある単語をキーワードとして、この単語間のリンクを辿ることにより、キーワードを含む文書を特定することができる。このＤｔｏＫインデックスのデータ構造の詳細については後述する。

【００２８】

集計処理部１３０は、文書ＤＢ２００に格納されている文書のうち、集計処理の対象として指定された文書（以下、指定文書と呼ぶ）を対象として、指定文書全体における出現頻度の高い単語を特定し、集計する。具体的には、集計処理部１３０は、出現頻度が予め定められた上位ｋ個に含まれる単語を集計する。本実施形態の集計処理部１３０は、次の何れかの手法を用いて集計処理を行う。

【００２９】

すなわち、一つは、集計対象（集計される属性）の単語ごとに、その単語を含む文書の数を調べ、最も多くの文書に含まれる単語から降順にｋ個の単語を選択する処理である。以下、この手法を第１手法と呼ぶ。より一般的に言い換えれば、第１手法は、属性ごとに、その属性を有する対象データを調べ、対象データとの関係が所定の基準を満たす属性を集計する手法である。

【００３０】

他の一つは、指定文書ごとに、その指定文書に含まれる単語を調べ、最も多くの指定文書に含まれる単語から降順にｋ個の単語を選択する処理である。以下、この手法を第２手法と呼ぶ。より一般的に言い換えれば、第２手法は、対象データごとに、その対象データが有する属性を調べ、対象データとの関係が所定の基準を満たす属性を集計する手法である。

【００３１】

本実施形態において、第１手法の処理は、ＤｔｏＫインデックスの単語間に設けられたリンクを辿ることによって行われる。集計処理部１３０の動作の詳細については後述する。

【００３２】
　また、集計処理部１３０は、出力候補保持部１３１を有している。出力候補保持部１３１には、集計処理部１３０による集計処理の過程で取得される、集計結果として出力する単語の候補が保持される。すなわち、集計処理部１３０は、第１手法または第２手法による処理において、集計対象となる個々の単語に対して、集計結果として出力する単語に該当するか否かを順次判断する。そして、集計結果として出力する単語に該当すると判断した単語を、出力候補保持部１３１に保持させる。言い換えれば、出力候補保持部１３１に保持される単語は、集計処理部１３０による集計処理の実行途中における、既に処理の済んだ単語である。集計処理が完了した後、集計処理部１３０は、出力候補保持部１３１に保持されている単語を、集計結果として出力する。
【００３３】
　判定部１４０は、集計処理部１３０による集計処理を第１手法と第２手法のどちらで行うかを判定する。上記の第１手法による処理は、集計対象である単語の種類が多い場合、多大な処理時間を要する。一方、第２手法による処理は、集計処理の対象である文書の数が多い場合、多大な処理時間を要する。そこで、判定部１４０は、集計処理部１３０による集計処理が効率良く実行されるように、第１手法と第２手法のどちらで集計処理を行うかを決定する。なお、具体的な決定方法は、特に限定しない。例えば、集計処理の対象である文書の数が予め設定された閾値よりも多い場合に第１手法により集計処理を実行すると決定し、文書の数が閾値以下である場合に第２手法により集計処理を実行すると決定しても良い。また、第１手法で集計処理を行う場合に要する処理時間と、第２手法で集計処理を行う場合に要する処理時間とを予想し、予想される処理時間の短い方の手法により集計処理を行うと決定しても良い。
【００３４】
　インデックス更新部１５０は、インデックス記憶部１２０に記憶されているインデックスを更新する。具体的には、インデックス更新部１５０は、文書ＤＢ２００の更新（文書の追加または削除）に応じて、追加された文書に対応するＤｔｏＫインデックスを追加したり、削除したりする。また、インデックス更新部１５０は、ＤｔｏＫインデックスの更新に伴い、ＤｔｏＫインデックスを管理するためのＤｔｏＫリスト（後述）も更新する。また、インデックス更新部１５０は、新たにＤｔｏＫインデックスを追加する更新処理を繰り返すことにより、インデックス記憶部１２０に記憶されるＤｔｏＫインデックス群を作成する。インデックス更新部１５０による更新処理の詳細については後述する。
【００３５】
＜インデックスの構成＞
　図２は、本実施形態のインデックス記憶部１２０に記憶されるインデックスの構成例を示す図である。図２に示すように、本実施形態では、単語リスト１２１と、文書ＤＢ２００に格納されている文書ごとのＤｔｏＫインデックス１２２とが用意される。言い換えれば、このＤｔｏＫインデックス１２２は、文書（対象データ）ごとに作成された、文書に含まれる単語（対象データが有する属性）のリスト（第１リスト）である。なお、図２に示す例では、ｎ個の文書に対するＤｔｏＫインデックス１２２が存在し、最後に文書ＤＢ２００に格納された文書（最も新しい文書）から順に、（１）～（ｎ）の枝番を付している。すなわち、符号１２２（ｎ）は、ｎ個の文書の中で最初に文書ＤＢ２００に格納された文書（最も古い文書）に対するＤｔｏＫインデックス１２２であり、符号１２２（１）は、最後に文書ＤＢ２００に格納された文書に対するＤｔｏＫインデックス１２２である。すなわち、ＤｔｏＫインデックス１２２は、最初に作成されたＤｔｏＫインデックス１２２を最後尾として、作成された順に並ぶ。
【００３６】
　また、特に図示しないが、インデックス記憶部１２０は、ＤｔｏＫインデックス１２２を図２に示す枝番の順番で管理するためのＤｔｏＫリストを記憶している。このＤｔｏＫリストは、ＤｔｏＫインデックス１２２の識別情報ｄｉ（ｉ：１≦ｉ≦ｎ）を添え字「ｉ」の昇順に並べて登録している。したがって、ＤｔｏＫリストにおいて、先頭のＤｔｏＫ

インデックス１２２の識別情報はｄ１であり、最後のＤｔｏＫインデックス１２２の識別情報はｄｎである。また、識別情報ｄｉの添え字「ｉ」は、図２に示したＤｔｏＫインデックス１２２の枝番（１）～（ｎ）に対応している。集計処理部１３０は、ＤｔｏＫリストから各ＤｔｏＫインデックス１２２（１）～１２２（ｎ）へランダム・アクセスが可能である。また、ＤｔｏＫリストは、リストの先頭と終端に対して、エントリ（要素）の追加および削除が可能なリスト構造である。このようなリストは、例えば、Ｊａｖａ（登録商標）のjava.util.LinkedListにより実現される。

【００３７】

単語リスト１２１は、文書ＤＢ２００に格納されている文書に含まれる全ての単語の識別情報のリスト（第２リスト）である。この単語リスト１２１には、各単語に関して、その単語の識別情報、その単語を含む文書の数（以下、出現文書数と呼ぶ）、その単語に関するリンク先の情報が登録される。リンク先の情報については後述する。また、単語リスト１２１は、各単語の識別情報に関してランダム・アクセスが可能であり、かつ、出現文書数に関してソート順を保持するマップ構造である。このようなリスト（マップ）は、例えば、Ｊａｖａのjava.util.TreeMapにより実現される。

【００３８】

図２に示す例では、単語リスト１２１には、ｍ個（識別情報「１」～「ｍ」）の単語が収録されている。単語リスト１２１に収録された単語は、最も多くの文書に含まれる単語から降順に並べられている。言い換えれば、単語リスト１２１において、対象データの属性である各単語は、対象データである文書全体における個々の文書に出現する頻度の降順にソートされている。なお、文書ＤＢ２００が更新されて文書が追加された場合、追加された文書にのみ含まれる単語が存在するならば、その単語に関するエントリが、単語リストの末尾に追加される。すなわち、図２に示す単語リスト１２１においては、識別情報「ｍ」のエントリの後に、識別情報「ｍ＋１」、「ｍ＋２」、……、というようにエントリが追加されていく。

【００３９】

ＤｔｏＫインデックス１２２は、文書ＤＢ２００に格納されている文書ごとに作成された、個々の文書に含まれる単語の情報を参照するインデックスである。ＤｔｏＫインデックス１２２には、対応する文書に含まれる各単語に関して、その単語の識別情報と、その単語に付与された点数と、その単語に関するリンク先の情報が登録される。リンク先の情報については後述する。単語に付与される点数は、任意に定義し、設定して良い。例えば、ＴＦ－ＩＤＦ（Term Frequency － Inverse Document Frequency）による重みの値などを点数として用いることができる。本実施形態では、各文書における、その単語の出現数を点数とする。また、ＤｔｏＫインデックス１２２に収録される単語は、この点数の値が最も大きい単語から降順に並べられている。

【００４０】

本実施形態において、単語リスト１２１および各ＤｔｏＫインデックス１２２（１）～１２２（ｎ）に登録されている各単語は、同一の単語どうしの間にリンクが張られている。本実施形態では、単語リスト１２１に登録されている識別情報を始点として、ＤｔｏＫインデックス１２２の枝番の順（すなわち、新しい文書から古い文書へ向かう順）にしたがってリンクが張られる。言い換えれば、属性である単語の間のリンクは、対象データである文書ごとのＤｔｏＫインデックス１２２（１）～１２２（ｎ）の順序に基づく特定の順序で張られる。また、枝番で次のＤｔｏＫインデックス１２２に同一の単語が無い場合は、枝番がより後のＤｔｏＫインデックス１２２であって同一の単語を含むもののうち、直近のＤｔｏＫインデックス１２２に含まれる単語に対してリンクが張られる。

【００４１】

具体的に、識別情報「１」の単語（以下、単語「１」と記す）について、図２を参照して設定されるリンクを説明する。単語「１」のリンクは、まず、単語リスト１２１の単語「１」から、ＤｔｏＫインデックス１２２（１）の単語「１」へ張られている。そして、ＤｔｏＫインデックス１２２（１）の単語「１」から、ＤｔｏＫインデックス１２２（２

）の単語「１」へ張られている。以下、図示は省略するが、ＤｔｏＫインデックス１２２の枝番の順にしたがって、単語「１」間のリンクが張られている。

【００４２】

ただし、ある文書が単語「１」を含まない場合、その文書のＤｔｏＫインデックス１２２を飛ばして、より後方のＤｔｏＫインデックス１２２に対してリンクが設定される。例えば、図示しないＤｔｏＫインデックス１２２（３）に対応する文書に単語「１」が含まれていない場合を考える。この場合、ＤｔｏＫインデックス１２２（２）の単語「１」からのリンクは、ＤｔｏＫインデックス１２２（３）を飛ばして、ＤｔｏＫインデックス１２２（４）の単語「１」へ張られる。同様に、ＤｔｏＫインデックス１２２（４）に対応する文書にも単語「１」が含まれていない場合、ＤｔｏＫインデックス１２２（２）の単語「１」からのリンクは、ＤｔｏＫインデックス１２２（５）の単語「１」へ張られる。

【００４３】

図２に示す例では、識別情報「２」の単語は、ＤｔｏＫインデックス１２２（１）に対応する文書に含まれていない。したがって、識別情報「２」の単語（以下、単語「２」と記す）のリンクは、ＤｔｏＫインデックス１２２（１）を飛ばして、単語リスト１２１の単語「２」からＤｔｏＫインデックス１２２（２）の単語「２」へ張られている。

【００４４】

したがって、本実施形態によるインデックスの単語間のリンクにおいて、ＤｔｏＫインデックス１２２（ｎ）に含まれる各単語は、その単語に関するリンクの終端となる。そして、ＤｔｏＫインデックス１２２（ｎ）に含まれていない各単語に関しては、ＤｔｏＫインデックス１２２（ｎ）よりも前方のＤｔｏＫインデックス１２２（１）～１２２（ｎ－１）の何れかにリンクの終端が存在する。

【００４５】

本実施形態によるインデックスの単語間のリンクは、既存の種々の手段により設定して良い。一例としては、単語リスト１２１および各ＤｔｏＫインデックス１２２（１）～１２２（ｎ）における各単語のエントリに、リンク先のＤｔｏＫインデックス１２２およびエントリを指定する情報を記述することによって実現される。他の一例としては、単語リスト１２１および各ＤｔｏＫインデックス１２２（１）～１２２（ｎ）における各単語のエントリに、他の文書のＤｔｏＫインデックス１２２に含まれている同一の単語のエントリへのポインタを記述することによって実現される。

【００４６】

＜集計処理部の動作＞

集計処理部１３０は、集計処理の対象として指定された指定文書全体における出現頻度の高い単語を特定し集計する処理を、第１手法または第２手法の一方により行う。第１手法は、集計対象（集計される属性）の単語ごとに、その単語を含む指定文書の数を調べ、最も多くの指定文書に含まれる単語から降順にｋ個の単語を選択する処理である。また、第２手法は、指定文書ごとに、その文書に含まれる単語を調べ、最も多くの文書に含まれる単語から降順にｋ個の単語を選択する処理である。

【００４７】

ここで、第２手法は、従来のＤｔｏＫインデックスを用いた既存の集計処理と同様である。本実施形態においても、図２を参照して説明したＤｔｏＫインデックス１２２を用いて従来と同様の処理を行うことができる。すなわち、集計処理部１３０は、まず、指定文書のＤｔｏＫインデックス１２２に基づいて、各指定文書に含まれる単語のリストを作成する。そして、集計処理部１３０は、作成したリストに基づき、最も多くの文書に含まれる単語から降順にｋ個の単語を選択する。

【００４８】

一方、第１手法は、従来技術におけるＫｔｏＤインデックスを用いた既存の集計処理と同様の考え方による処理である。ただし、本実施形態では、ＫｔｏＤインデックスを用いず、図２を参照して説明したＤｔｏＫインデックス１２２における単語間のリンクを利用して処理を行う。

【００４９】
図３および図４は、第１の手法により集計処理を行う場合の集計処理部１３０の動作を示すフローチャートである。集計処理部１３０は、単語リスト１２１に収録された単語に対する処理を、例えば、識別情報「１」の単語から順に処理対象として着目して行う。すなわち、集計処理部１３０は、まず、未処理の単語が残っているか否かを判断する（Ｓ３０１）。そして、未処理の単語がある場合は（Ｓ３０１でＹｅｓ）、集計処理部１３０は、未処理の単語から処理対象の単語（以下、対象単語と呼ぶ）を選択する（Ｓ３０２）。ここで、対象単語として選択される単語は、例えば、未処理の単語のうちで識別情報の値が最も小さい単語である。単語リスト１２１に収録された単語は、最も多くの文書に含まれる単語から降順に並べられている。したがって、識別情報の値が最も小さい単語は、未処理の単語のうちで最も多くの文書に含まれる単語である。

【００５０】
次に、集計処理部１３０は、出力候補保持部１３１に保持されている処理済みの単語（以下、処理済み単語と呼ぶ）の数が集計数のｋ個か否かを判断する（Ｓ３０３）。そして、処理済み単語の数がｋ個であれば（Ｓ３０３でＹｅｓ）、集計処理部１３０は、対象単語の出現文書数「ｇ」を取得する（Ｓ３０４）。この出現文書数「ｇ」は、文書ＤＢ２００に格納されている文書全体のうち、対象単語を含む文書の数である。

【００５１】
次に、集計処理部１３０は、対象単語の出現文書数「ｇ」と、出力候補保持部１３１に保持されている各処理済み単語の出現文書数「ｈ１」とを比較する（Ｓ３０５）。ここで、処理済み単語の出現文書数「ｈ１」は、その処理済み単語を含む指定文書の数である。すなわち、処理済み単語の出現文書数「ｈ１」は、文書ＤＢ２００に格納されている文書全体のうち、その処理済み単語を含み、かつ、集計処理の対象として指定された文書の数である。後述するように、処理済み単語の出現文書数は、出力候補保持部１３１に保持されている。対象単語の出現文書数「ｇ」が何れかの処理済み単語の出現文書数「ｈ１」よりも大きい場合（Ｓ３０６でＹｅｓ）、集計処理部１３０は、この対象単語に関して、ＤｔｏＫインデックス１２２のリンクを辿って、この対象単語を含む文書のリストを取得する（Ｓ３０７）。

【００５２】
次に、集計処理部１３０は、取得したリストに含まれる文書のうち、絞り込み条件を満足する文書（指定文書）の数を算出する（Ｓ３０８）。そして、集計処理部１３０は、算出された指定文書の数「ｈ２」と、各処理済み単語の出現文書数「ｈ１」とを比較する（Ｓ３０９）。指定文書数「ｈ２」が何れかの処理済み単語の出現文書数「ｈ１」よりも大きい場合（Ｓ３１０でＹｅｓ）、集計処理部１３０は、対象単語および処理済み単語のうち、出現文書数の多い方からｋ個を選択する。そして、集計処理部１３０は、選択した各単語と、その単語の出現文書数とを対応付けて、出力候補保持部１３１に記憶させる。これにより、集計結果として出力する単語の候補である処理済み単語が更新される（Ｓ３１１）。なお、処理済み単語の数の上限はｋ個なので、今回の処理の対象単語が処理済み単語に追加されたことに伴い、更新前の処理済み単語のうち、最も出現文書数の少ない単語の情報が出力候補保持部１３１から消去される。この後、集計処理部１３０の処理は、Ｓ３０１に戻る。そして、未処理の単語が残っていれば（Ｓ３０１でＹｅｓ）、Ｓ３０２以降の処理が行われる。

【００５３】
全ての処理済み単語の出現文書数「ｈ１」が指定文書数「ｈ２」以上であった場合（Ｓ３１０でＮｏ）、今回の処理の対象単語は、処理済み単語に追加されない。したがって、出力候補保持部１３１に記憶された処理済み単語は更新されることなく、集計処理部１３０の処理は、Ｓ３０１に戻る。そして、未処理の単語が残っていれば（Ｓ３０１でＹｅｓ）、Ｓ３０２以降の処理が行われる。

【００５４】
ここで、Ｓ３０３において、処理済み単語の数が集計数ｋ個に達していない場合を考え

る（Ｓ３０３でＮｏ）。この場合、今回の処理の対象単語は、必ず、集計結果として出力する単語の候補となる。したがって、集計処理部１３０は、この対象単語に関して、ＤｔｏＫインデックス１２２のリンクを辿って、この対象単語を含む文書のリストを取得する（Ｓ３１３）。そして、集計処理部１３０は、取得したリストに含まれる文書のうち、絞り込み条件を満足する文書（指定文書）の数を算出する（Ｓ３１４）。この後、集計処理部１３０は、対象単語と算出した指定文書の数とを対応付けて、出力候補保持部１３１に記憶させる。これにより、集計結果として出力する単語の候補である処理済み単語が更新される（Ｓ３１１）。なお、この場合は、更新後の処理済み単語の数はｋ個以下なので、何れの処理済み単語も、出力候補保持部１３１から消去されない。この後、集計処理部１３０の処理は、Ｓ３０１に戻る。そして、未処理の単語が残っていれば（Ｓ３０１でＹｅｓ）、Ｓ３０２以降の処理が行われる。

【００５５】

一方、Ｓ３０１において、未処理の単語が残っていなければ（Ｓ３０１でＮｏ）、全ての単語に対して上記の処理が済んだので、集計処理部１３０は、出力候補保持部１３１に記憶されているｋ個の処理済み単語を、集計結果として出力し（Ｓ３１２）、処理を終了する。

【００５６】

また、Ｓ３０６において、全ての処理済み単語の出現文書数「ｈ１」が対象単語の出現文書数「ｇ」以上であった場合を考える（Ｓ３０６でＮｏ）。この場合、単語リスト１２１に収録された単語は、最も多くの文書に含まれる単語から降順に並べられているため、今回よりも後の処理において対象単語となる単語の出現文書数が何れかの処理済み単語の出現文書数「ｈ１」よりも大きくなることはない。すなわち、今回よりも後の処理において処理済み単語が更新されることはない。したがって、集計処理部１３０は、出力候補保持部１３１に記憶されているｋ個の処理済み単語を、集計結果として出力し（Ｓ３１２）、処理を終了する。

【００５７】

このように、Ｓ３０５、Ｓ３０６に示す処理によれば、単語リスト１２１に収録された全ての単語に対して処理を行っていない段階でも、処理済み単語が更新される可能性が無くなった場合に、集計処理を終了する。これにより、Ｓ３０１～Ｓ３１１の処理の繰り返し回数を、単語リスト１２１に収録された単語の数よりも少なく抑えることができる。

【００５８】

以上のように、本実施形態は、収録した単語間にリンクを設定したＤｔｏＫインデックス１２２を用いて、第１手法による処理および第２手法による処理のどちらも実行できる。したがって、第１手法による処理を行うためにＫｔｏＤインデックスを用意する場合と比較して、インデックスを保持するために必要なメモリ領域（記憶容量）を小さく抑えることができる。

【００５９】

なお、単にインデックスを保持するために必要なメモリ領域（記憶容量）を削減するのであれば、適当な圧縮方式によりインデックスをデータ圧縮してデータサイズを小さくすることが可能である。しかし、文書ＤＢ２００の更新（文書の追加または削除）に応じてデータ圧縮されたインデックスを更新することは困難である。そのため、文書ＤＢ２００が更新された場合は、インデックス自体を作成し直し、改めてデータ圧縮することが必要となる。これに対し、本実施形態では、限定的ではあるが、後述のように、文書ＤＢ２００の更新に応じてＤｔｏＫインデックス１２２を更新可能としながら、上記のようにインデックスを保持するために必要なメモリ領域（記憶容量）の削減を実現している。

【００６０】

＜ＤｔｏＫインデックスの作成および更新＞

本実施形態において、文書ＤＢ２００が更新されると、これに伴って、インデックス記憶部１２０に記憶された単語リスト１２１およびＤｔｏＫインデックス１２２も更新される。単語リスト１２１に関しては、文書の追加または削除により、収録している各単語を

含む文書の数が変わる。そこで、本実施形態の集計システム１００のインデックス更新部１５０は、例えば、文書ＤＢ２００が更新される度に、単語リスト１２１に収録している単語をソートし直す。また、本実施形態のインデックス更新部１５０は、追加された文書にのみ含まれる単語が存在するならば、その単語の識別情報を、単語リストの末尾に追加することにより、単語リスト１２１を更新する。

【００６１】

次に、ＤｔｏＫインデックス１２２の作成および更新について説明する。ＤｔｏＫインデックス１２２の作成は、一つの文書に対応する新たなＤｔｏＫインデックス１２２を追加する更新処理を、文書ＤＢ２００に格納されたｎ個の文書に対して実行することにより行われる。

【００６２】

図５は、ＤｔｏＫインデックス１２２の作成処理の手順を示すフローチャートである。図５に示すように、インデックス更新部１５０は、まず、文書ＤＢ２００に格納されている文書のうち、未処理の（ＤｔｏＫインデックス１２２の作成が行われていない）文書があるか否かを調べる（Ｓ４０１）。未処理の文書があれば（Ｓ４０１でＹｅｓ）、インデックス更新部１５０は、未処理の文書の一つを取得する（Ｓ４０２）。そして、インデックス更新部１５０は、取得した文書を処理対象として、新規文書追加処理を行う（Ｓ４０３）。以上の処理を文書ＤＢ２００に格納されている各文書に対して行い、未処理の文書が無くなったならば（Ｓ４０１でＮｏ）、インデックス更新部１５０は、処理を終了する。

【００６３】

図６は、新規文書追加処理の手順を示すフローチャートである。図６に示すように、インデックス更新部１５０は、まず、処理対象として取得した一つの文書に対応するＤｔｏＫインデックス１２２を作成する（Ｓ５０１）。以下、作成されたＤｔｏＫインデックス１２２を、ＤｔｏＫリストに登録される各ＤｔｏＫインデックス１２２の識別情報ｄｉを用いてインデックスｄｉと記載する（図６のＳ５０１でもインデックスｄｉと記載）。また、インデックスｄｉに対応する文書を文書ｄｉと記載する。インデックス更新部１５０は、作成したインデックスｄｉをＤｔｏＫリストの先頭に追加する（Ｓ５０２）。

【００６４】

次に、インデックス更新部１５０は、インデックスｄｉに登録されている単語（文書ｄｉに含まれる単語）のうち、未処理の（リンクの設定が更新されていない）単語が有るか否かを調べる（Ｓ５０３）。未処理の単語が有る場合（Ｓ５０３でＹｅｓ）、インデックス更新部１５０は、未処理の単語の一つを選択し（Ｓ５０４）、選択した単語が単語リスト１２１に登録されているか否かを調べる（Ｓ５０５）。そして、選択した単語が登録されているならば（Ｓ５０５でＹｅｓ）、インデックス更新部１５０は、単語リスト１２１におけるその単語のエントリに登録されているリンク先の情報を、処理中のインデックスｄｉにおけるその単語のエントリを指すように更新する（Ｓ５０７）。

【００６５】

一方、選択した単語が単語リスト１２１に登録されていない場合（Ｓ５０５でＮｏ）、インデックス更新部１５０は、選択した単語のエントリを単語リスト１２１に追加する（Ｓ５０６）。そして、インデックス更新部１５０は、単語リスト１２１において作成したエントリのリンク先の情報を、処理中のインデックスｄｉにおけるその単語のエントリを指すように更新する（Ｓ５０７）。

【００６６】

次に、インデックス更新部１５０は、インデックスｄｉにおけるその単語のエントリに登録されているリンク先の情報を、ＤｔｏＫリストにおいて後方かつ直近の他のＤｔｏＫインデックス１２２におけるその単語のエントリを指すように更新する（Ｓ５０８）。

【００６７】

インデックス更新部１５０は、以上のＳ５０３～Ｓ５０８の処理を、インデックスｄｉに含まれる各単語に対して行い、未処理の単語が無くなったならば（Ｓ５０３でＮｏ）、

インデックス更新部１５０は、処理を終了する。
【００６８】
以上、ＤｔｏＫインデックス１２２を追加する処理について説明した。次に、インデックス記憶部１２０に記憶されているＤｔｏＫインデックス１２２を削除する処理について説明する。図２を参照して説明した本実施形態のＤｔｏＫインデックス１２２は、単語間にリンクが設定されている。そのため、ＤｔｏＫインデックス１２２を削除する場合は、削除対象のＤｔｏＫインデックス１２２における単語のリンクを設定し直す必要がある。したがって、任意のＤｔｏＫインデックス１２２を削除することは容易ではない。しかし、ＤｔｏＫリストの終端に対応する（すなわち、最も古い）ＤｔｏＫインデックス１２２（識別情報ｄｎ）については、削除することが容易である。これは、ＤｔｏＫインデックス１２２の全ての単語が、その単語に関するリンクの終端だからである。この場合、インデックス記憶部１２０から識別情報ｄｎのＤｔｏＫインデックス１２２（図２に示す例では、ＤｔｏＫインデックス１２２（ｎ））を削除し、ＤｔｏＫリストから終端のエントリを削除すれば良い。
【００６９】
＜インデックスの実装例＞
図７および図８は、Ｊａｖａプログラミング言語を用いた、本実施形態によるＤｔｏＫインデックス１２２の実装例を示す図である。図７は、コードの例を示す図であり、図７（ａ）はＤｔｏＫインデックス１２２のエントリを設定するコード、図７（ｂ）はＤｔｏＫリストを生成するコード、図７（ｃ）は新しいＤｔｏＫインデックス１２２を追加するコード、図７（ｄ）はＤｔｏＫインデックス１２２を削除するコードである。図８は、図７に示すコードにより設定されたリンクの様子を示す図である。
【００７０】
図７に示すように実装した場合、ＤｔｏＫインデックス１２２の単語（エントリ）間のリンクは、リンク先のＤｔｏＫインデックス１２２およびエントリの位置を数値で指定することにより実現される。図７（ａ）に示す例では、リンク先のＤｔｏＫインデックス１２２は、「nextDocId」の値で指定される。ここで、ＤｏｃＩｄとは、後述の文書ＩＤである。また、リンク先のエントリの位置は、「nextEntryIndex」の値で指定される。また、「keywordId」は、各単語の識別情報である。そして、「score」は、各単語に付与された点数である。また、図７（ｂ）に示すように、Ｊａｖａの標準ライブラリに存在するクラスである「java.util.TreeMap」により、ＤｔｏＫリスト「DLList」が生成される。このＤｔｏＫリスト「DLList」は、ＤｔｏＫインデックス１２２（図７ではＤＬ［ｉ］と記載）の全体を識別情報ｄｉの昇順（ＤＬ［１］～ＤＬ［ｎ］）に格納する。
【００７１】
また、図７（ｃ）に示す例では、ＤｔｏＫインデックス１２２が追加される場合に、追加されるＤｔｏＫインデックス１２２に対応する文書の識別情報（文書ＩＤ）が設定される。この文書ＩＤは、ＤｔｏＫリスト「DLList」に登録されるＤｔｏＫインデックス１２２の識別情報ｄｉとは別に設定される。例えば、それまでに使用した文書ＩＤの最大値を記憶しておき、その最大値よりも１大きい値を新たに設定する文書ＩＤの値とする。このようにすれば、図７（ｄ）に示すように、文書ＩＤが最小値であるＤｔｏＫインデックス１２２を削除することにより、ＤｔｏＫリスト「DLList」の終端のエントリに対応するＤｔｏＫインデックス１２２が削除される。
【００７２】
図８に示す例では、識別情報「２００」の単語に関して、ＤｔｏＫインデックス１２２（３）からＤｔｏＫインデックス１２２（５）へ設定されたリンクが示されている。図示の例において、ＤｔｏＫインデックス１２２（３）の文書ＩＤは「１０」であり、ＤｔｏＫインデックス１２２（５）の文書ＩＤは「６」である。また、ＤｔｏＫインデックス１２２（３）において、識別情報「２００」の単語は、２番目のエントリ「Entry[1]」に登録されている。ＤｔｏＫインデックス１２２（３）のエントリ「Entry[1]」を参照すると、識別情報「２００」の単語の点数は「１．５」である。また、リンク先は、文書ＩＤ「

６」のＤｔｏＫインデックス１２２のエントリ「Entry[2]」となっている。したがって、リンク先である、ＤｔｏＫインデックス１２２（５）の３番目のエントリ「Entry[2]」には、識別情報「２００」の単語が登録されている。

【００７３】

図８に示す例において、文書ＩＤ「６」の文書が削除された場合を考える。ＤｔｏＫリスト「DLList」およびＤｔｏＫインデックス１２２が更新されると、文書ＩＤ「６」のＤｔｏＫインデックス１２２（５）が削除される。これにより、ＤｔｏＫリスト「DLList」に文書ＩＤ「６」のＤｔｏＫインデックス１２２（５）が存在しなくなる。そのため、ＤｔｏＫインデックス１２２（３）のエントリ「Entry[1]」からＤｔｏＫインデックス１２２（５）のエントリ「Entry[2]」へのリンクを辿れなくなる。そして、識別情報「２００」の単語に関するリンクは、ＤｔｏＫインデックス１２２（３）のエントリ「Entry[1]」が終端となる。

【００７４】

＜インデックスの他の構成および実装例＞

以上の実施形態において、単語リスト１２１および各ＤｔｏＫインデックス１２２（１）～１２２（ｎ）に登録されている単語間のリンクは、単語リスト１２１に登録されている識別情報を始点として、識別情報ｄｉの順にしたがって設定した。これに対し、ＤｔｏＫインデックス１２２の実装によっては、識別情報ｄｉの降順（すなわち、古い文書から新しい文書へ向かう順）にリンクを設定することが望ましい場合がある。

【００７５】

図９および図１０は、Ｊａｖａプログラミング言語を用いた、本実施形態によるＤｔｏＫインデックス１２２の他の実装例を示す図である。図９は、ＤｔｏＫインデックス１２２のエントリを設定するコードの例を示す図である。図１０は、図９に示すコードにより設定されたリンクの様子を示す図である。なお、ＤｔｏＫリストを生成するコード、新しいＤｔｏＫインデックス１２２を追加するコード、ＤｔｏＫインデックス１２２を削除するコードは、図７（ｂ）～（ｄ）に示したコードと同様とする。

【００７６】

図９に示すように実装した場合、ＤｔｏＫインデックス１２２の単語（エントリ）間のリンクは、リンク先のエントリをポインタにより直接指し示すことにより実現される。図９に示す例では、リンク先のＤｔｏＫインデックス１２２のエントリが、ＪａｖａやＣ／Ｃ＋＋等のプログラミング言語でサポートされているポインタ（または同等の機能）で指定される。図示の例では、「Entry next」にポインタが記録される。このように、ポインタを用いてリンクが設定される場合、図７および図８に示した実装例とは反対の向きにリンクを設定する。すなわち、ある単語に関するリンクは、その単語が出現する最も古い（識別情報ｄｉの値が最も大きい）ＤｔｏＫインデックス１２２のエントリを始点とし、単語リスト１２１におけるその単語のエントリを終端とする。

【００７７】

また、この実装例では、単語リスト１２１のエントリから、各単語が出現する最も古いＤｔｏＫインデックス１２２のエントリへ向かうリンク（以下、逆向きリンクと呼ぶ）も設定される。この逆向きリンクは、集計処理部１３０が第１手法による集計処理を行う場合に、単語のリンクを辿るために用いられる。すなわち、集計処理部１３０は、逆向きリンクにより、通常のリンクの始点を見つける。

【００７８】

図１０に示す例では、識別情報「２００」の単語に関して、ＤｔｏＫインデックス１２２（５）からＤｔｏＫインデックス１２２（３）へ、図８に示した例とは反対の向きに設定されたリンクが示されている。図示の例において、ＤｔｏＫインデックス１２２（３）の文書ＩＤは「１０」であり、ＤｔｏＫインデックス１２２（５）の文書ＩＤは「６」である。また、ＤｔｏＫインデックス１２２（５）において、識別情報「２００」の単語は、３番目のエントリ「Entry[2]」に登録されている。ＤｔｏＫインデックス１２２（５）のエントリ「Entry[2]」を参照すると、識別情報「２００」の単語の点数は「１．５」で

ある。そして、リンク先（ＤｔｏＫインデックス１２２（３）のエントリ「Entry[1]」）へのポインタが設定されている。したがって、リンク先である、ＤｔｏＫインデックス１２２（３）のエントリ「Entry[1]」には、識別情報「２００」の単語が登録されている。

【００７９】

図１１は、図９および図１０に示す実装例によるインデックスの構成を示す図である。図１１に示す構成例は、図２に示した構成例と同様に、単語リスト１２１と、文書ＤＢ２００に格納されている文書ごとのＤｔｏＫインデックス１２２とを含む。そして、ｎ個の文書に対するＤｔｏＫインデックス１２２が存在し、最後に文書ＤＢ２００に格納された文書（最も新しい文書）から順に、（１）～（ｎ）の枝番を付している。

【００８０】

図１１に示すインデックス構成では、単語ごとに、その単語が出現する最も古いＤｔｏＫインデックス１２２のエントリから単語リスト１２１におけるその単語のエントリへ至る通常のリンクと、逆向きリンクとが張られている。図１１においては、通常のリンクを実線で示し、逆向きリンクを破線で示している。

【００８１】

具体的に、識別情報「１」の単語（単語「１」）について、図１１に示すリンクを説明する。単語「１」の通常のリンクは、まず、単語「１」が出現する最も古いＤｔｏＫインデックス１２２（ｎ）のエントリから、ＤｔｏＫインデックス１２２（ｎ－１）のエントリへ張られている。以下、図示は省略するが、単語「１」間のリンクは、ＤｔｏＫインデックス１２２の枝番の降順にしたがって張られ、ＤｔｏＫインデックス１２２（２）のエントリに至っている。そして、単語「１」のリンクは、ＤｔｏＫインデックス１２２（２）のエントリから、ＤｔｏＫインデックス１２２（１）のエントリを経て、単語リスト１２１における単語「１」のエントリを終端としている。

【００８２】

なお、単語「１」を含まない文書のＤｔｏＫインデックス１２２を飛ばしてリンクが張られることは、図２に示した構成例と同様である。例えば、図１１において、単語「２」のリンクは、ＤｔｏＫインデックス１２２（１）を飛ばして、ＤｔｏＫインデックス１２２（２）のエントリから単語リスト１２１における単語「２」のエントリへ張られている。すなわち、単語「２」は、ＤｔｏＫインデックス１２２（１）の文書には出現しない。また、単語「３」のリンクは、ＤｔｏＫインデックス１２２（ｎ）のエントリからＤｔｏＫインデックス１２２（２）のエントリへ張られている。すなわち、単語「３」は、図示しないＤｔｏＫインデックス１２２（３）からＤｔｏＫインデックス１２２（ｎ－２）の文書には出現しない。

【００８３】

また、図１１に示すインデックス構成において、ある単語に関する通常のリンクの始点は、その単語が出現する最も古いＤｔｏＫインデックス１２２であり、必ずしも全体で最も古いＤｔｏＫインデックス１１２（ｎ）ではない。例えば、図１１の例において、単語「２」に関するリンクを考える。単語「２」のリンクの始点は、ＤｔｏＫインデックス１２２（ｎ）およびＤｔｏＫインデックス１２２（ｎ－１）のどちらにも存在しない。したがって、単語「２」のリンクの始点は、ＤｔｏＫインデックス１２２（ｎ－１）よりも新しい何れかのＤｔｏＫインデックス１２２である。

【００８４】

また、単語「１」の逆向きリンクは、単語リスト１２１における単語「１」のエントリから、単語「１」が出現する最も古いＤｔｏＫインデックス１２２（ｎ）のエントリへ張られている。単語「３」の逆向きリンクも同様である。これに対し、単語「２」の逆向きリンクは、単語「２」が出現する最も古いＤｔｏＫインデックス１２２が図示されていないため、記載を省略している。すなわち、単語「２」が出現する最も古いＤｔｏＫインデックス１２２は、ＤｔｏＫインデックス１２２（３）～ＤｔｏＫインデックス１２２（ｎ－２）の何れかである。

【００８５】

図９～図１１を参照して説明した実装例において、新たなＤｔｏＫインデックス１２２を追加する更新については、リンクの方向が反対方向になることを除き、図６に示した手順と同様の手順で実行することができる。リンクの設定に関しては、図６のＳ５０５で、選択した単語が単語リスト１２１に登録されているならば、（Ｓ５０５でＹｅｓ）、インデックス更新部１５０は、追加されたＤｔｏＫインデックス１２２におけるその単語のエントリのポインタを、単語リスト１２１におけるその単語のエントリを指すように更新する。また、インデックス更新部１５０は、単語リスト１２１におけるその単語のエントリから逆向きリンクおよび通常のリンクを辿り、現在、単語リスト１２１におけるその単語のエントリを指すポインタを有するＤｔｏＫインデックス１２２のエントリを検出する。そして、インデックス更新部１５０は、検出したＤｔｏＫインデックス１２２のエントリのポインタを、追加したＤｔｏＫインデックス１２２におけるその単語のエントリを指すように更新する。

【００８６】

一方、図６のＳ５０５で、選択した単語が単語リスト１２１に登録されていない場合は（Ｓ５０５でＮｏ）、インデックス更新部１５０は、選択した単語のエントリを単語リスト１２１に追加する。そして、インデックス更新部１５０は、追加したＤｔｏＫインデックス１２２におけるその単語のエントリのポインタを、単語リスト１２１におけるその単語のエントリを指すように更新してリンクを設定する。また、インデックス更新部１５０は、単語リスト１２１におけるその単語のエントリのポインタを、追加したＤｔｏＫインデックス１２２におけるその単語のエントリを指すように更新して逆向きリンクを設定する。

【００８７】

これに対し、ＤｔｏＫインデックス１２２を削除する更新の場合、単にＤｔｏＫインデックス１２２を削除するだけでなく、削除対象のＤｔｏＫインデックス１２２に収録されている各単語に関して設定されたリンクを更新する。削除対象である最も古いＤｔｏＫインデックス１２２に収録されている各単語は、その単語に関する通常のリンクの始点であり、かつ、その単語に関する逆向きリンクのリンク先である。そのため、ＤｔｏＫインデックス１２２を削除した後に、これらのリンクおよび逆向きリンクを正しく辿れるように、リンクを更新することが必要である。

【００８８】

図１２は、図９～図１１の実装例において、ＤｔｏＫインデックス１２２を削除する処理の手順を示すフローチャートである。図１２に示すように、インデックス更新部１５０は、まず、削除対象のＤｔｏＫインデックス１２２（ｎ）に登録されている単語のうち、未処理の（リンクの設定が更新されていない）単語があるか否かを調べる（Ｓ１１０１）。未処理の単語がある場合（Ｓ１１０１でＹｅｓ）、インデックス更新部１５０は、未処理の単語の一つを選択し（Ｓ１１０２）、選択した単語に対して張られている逆向きリンクのリンク先を更新する（Ｓ１１０３）。

【００８９】

具体的には、インデックス更新部１５０は、削除対象のＤｔｏＫインデックス１２２（ｎ）における選択した単語のエントリに登録されている通常のリンクのリンク先へのポインタの情報を取得する。このポインタにより示されるリンク先のＤｔｏＫインデックス１２２は、削除対象のＤｔｏＫインデックス１２２（ｎ）が削除された後に、選択した単語が出現する最も古いＤｔｏＫインデックス１２２となる。そして、インデックス更新部１５０は、単語リスト１２１における選択した単語に関する逆向きリンクのリンク先へのポインタを、取得したポインタの情報に基づいて更新する。

【００９０】

インデックス更新部１５０は、以上のＳ１１０１～Ｓ１１０３の処理を、削除対象のＤｔｏＫインデックス１２２（ｎ）に登録されている各単語に対して行う。これにより、ＤｔｏＫインデックス１２２（ｎ）の各単語に対して設定されていた逆向きリンクが、ＤｔｏＫインデックス１２２（ｎ）の削除後に各単語が出現する最も古いＤｔｏＫインデック

ス１２２へ、それぞれ設定し直される。そして、未処理の単語が無くなったならば（Ｓ１１０１でＮｏ）、インデックス更新部１５０は、削除対象のＤｔｏＫインデックス１２２（ｎ）を削除して（Ｓ１１０４）、処理を終了する。

【００９１】

図１３は、図１１に示したインデックス構成において、ＤｔｏＫインデックス１２２（ｎ）を削除した様子を示す図である。図１３を参照すると、ＤｔｏＫインデックス１２２（ｎ）を削除したことにより、単語「１」に関する通常のリンクの始点（逆向きリンクのリンク先）は、ＤｔｏＫインデックス１２２（ｎ－１）における単語「１」のエントリとなっている。また、単語「２」に関する通常のリンクの始点（逆向きリンクのリンク先）は、ＤｔｏＫインデックス１２２（２）における単語「２」のエントリとなっている。

【００９２】

＜ハードウェア構成例＞

図１４は、本実施形態の集計システム１００を構成するのに好適なハードウェア構成例を示す図である。ここでは、コンピュータにより構成する場合について説明する。図１４に示すコンピュータは、演算手段であるＣＰＵ（Central Processing Unit）１０ａと、主記憶手段であるメモリ１０ｃを備える。また、外部デバイスとして、磁気ディスク装置（ＨＤＤ：Hard Disk Drive）１０ｇ、ネットワーク・インターフェイス１０ｆ、ディスプレイ装置を含む表示機構１０ｄ、キーボードやマウス等の入力デバイス１０ｉ等を備える。

【００９３】

図１４に示す構成例では、メモリ１０ｃおよび表示機構１０ｄは、システム・コントローラ１０ｂを介してＣＰＵ１０ａに接続されている。また、ネットワーク・インターフェイス１０ｆ、磁気ディスク装置１０ｇおよび入力デバイス１０ｉは、Ｉ／Ｏコントローラ１０ｅを介してシステム・コントローラ１０ｂと接続されている。各構成要素は、システム・バスや入出力バス等の各種のバスによって接続される。

【００９４】

図１４において、磁気ディスク装置１０ｇにはＯＳのプログラムやアプリケーション・プログラムが格納されている。そして、これらのプログラムがメモリ１０ｃに読み込まれてＣＰＵ１０ａに実行されることにより、集計システム１００における集計処理部１３０、判定部１４０およびインデックス更新部１５０の各機能が実現される。また、メモリ１０ｃや磁気ディスク装置１０ｇにより、インデックス記憶部１２０が実現される。また、入力デバイス１０ｉおよびプログラム制御されたＣＰＵ１０ａにより、入力受け付け部１１０が実現される。さらにまた、本実施形態では、文書ＤＢ２００を磁気ディスク装置１０ｇにより実現しても良い。なお、図１４は、本実施形態の集計システム１００を実現するのに好適なコンピュータのハードウェア構成を例示するに過ぎず、集計システム１００の具体的構成は、図１４に示す構成に限定されない。

【００９５】

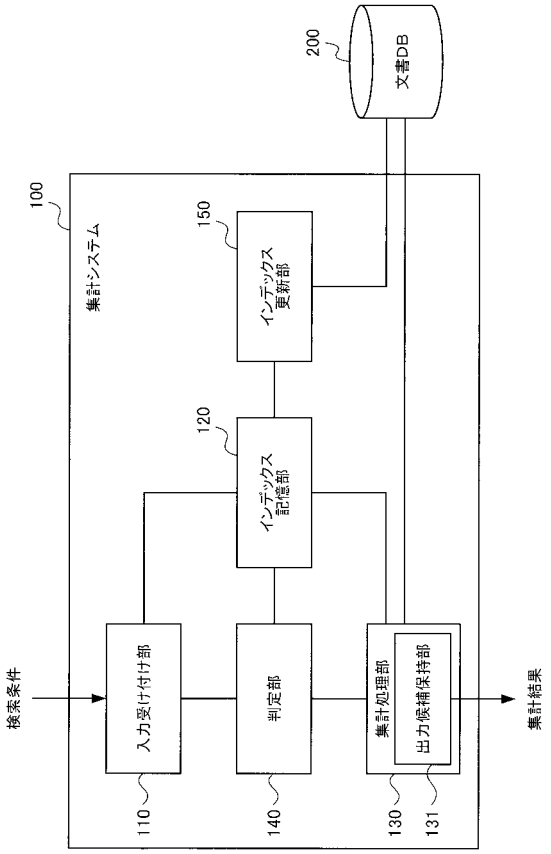
以上、本実施形態について説明したが、本実施形態は、上記の具体的構成に限定されるものではない。本実施形態は、複数種類の属性を持つ対象データの集合中に出現する属性を、何らかの規則に基づいて各属性に付与された点数の高いものから順に所定個数求める集計処理に対して、広く適用できるものである。すなわち、本実施形態が適用される集計処理における対象データは、上記の処理対象として絞り込まれた文書には限定されない。また、本実施形態が適用される集計処理における属性は、各文書に含まれる単語に限定されない。その他、上記の実施形態に、種々の変更または改良を加えたものも、本発明の技術的範囲に含まれる。

【符号の説明】

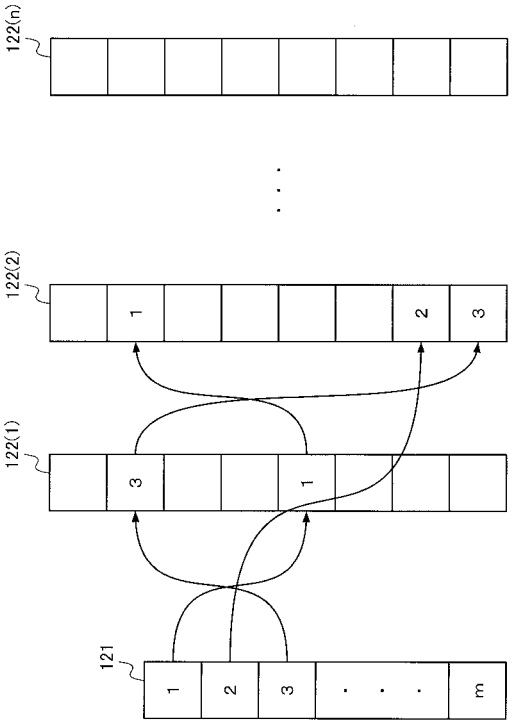
【００９６】

１００…集計システム、１１０…入力受け付け部、１２０…インデックス記憶部、１２１…単語リスト、１２２…ＤｔｏＫインデックス、１３０…集計処理部、１４０…判定部、１５０…インデックス更新部

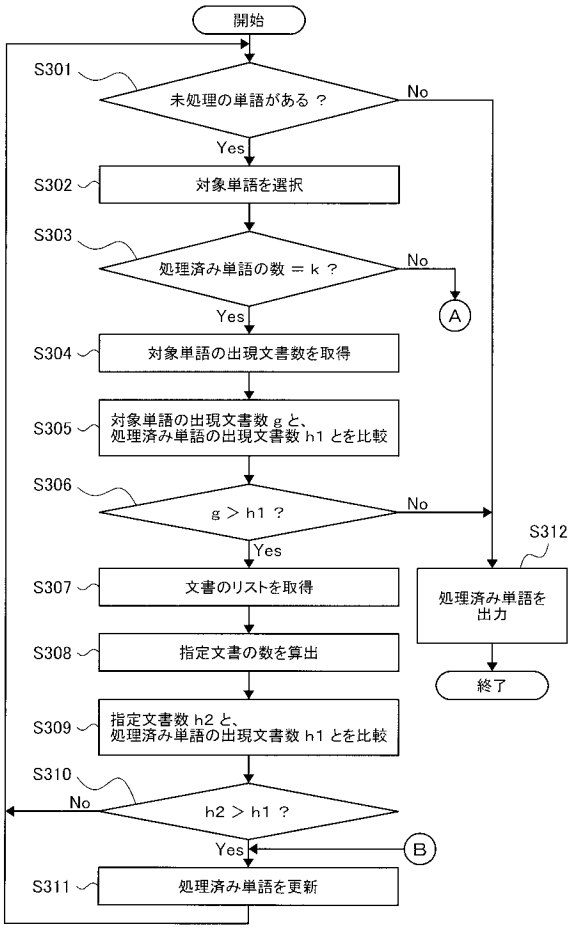
【図１】


200
文書DB
100
集計システム
150
インデックス更新部
120
インデックス記憶部
検索条件
入力受け付け部
判定部
集計処理部
出力候補保持部
集計結果
110
140
130
131

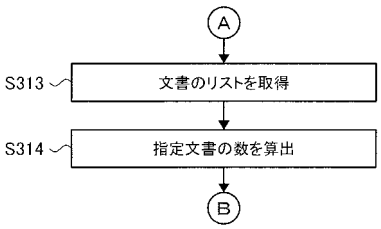
【図２】


122(n)
122(2)
1
2
3
122(1)
3
1
121
1
2
3
m

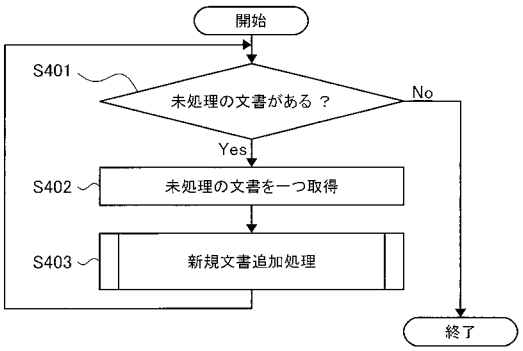
【図３】


開始
S301
未処理の単語がある？
No
Yes
S302
対象単語を選択
S303
処理済み単語の数 ＝ k ？
No
Yes
A
S304
対象単語の出現文書数を取得
S305
対象単語の出現文書数 g と、
処理済み単語の出現文書数 h1 とを比較
S306
g ＞ h1 ？
No
Yes
S312
処理済み単語を出力
終了
S307
文書のリストを取得
S308
指定文書の数を算出
S309
指定文書数 h2 と、
処理済み単語の出現文書数 h1 とを比較
S310
No
h2 ＞ h1 ？
Yes
B
S311
処理済み単語を更新

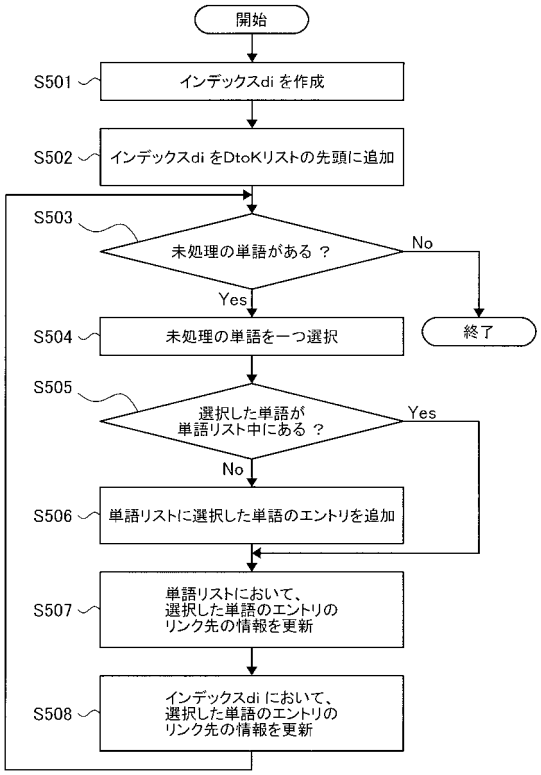
【図４】


A
S313
文書のリストを取得
S314
指定文書の数を算出
B

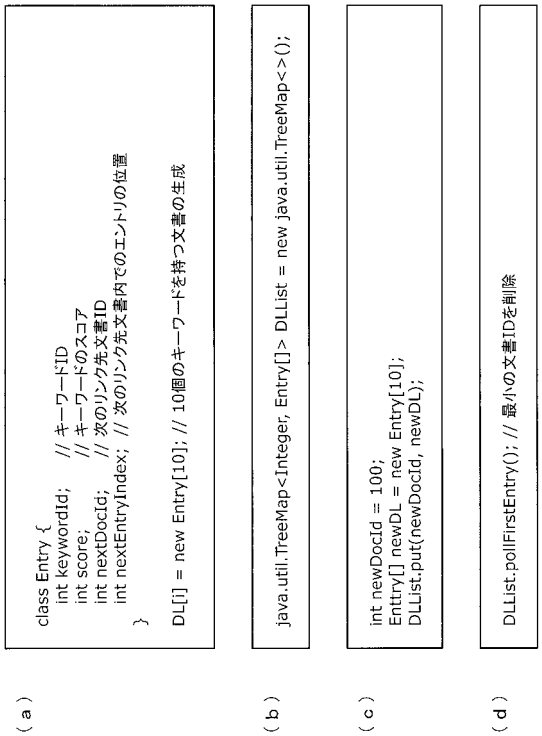
【図５】


開始
S401
未処理の文書がある ？
No
Yes
S402
未処理の文書を一つ取得
S403
新規文書追加処理
終了

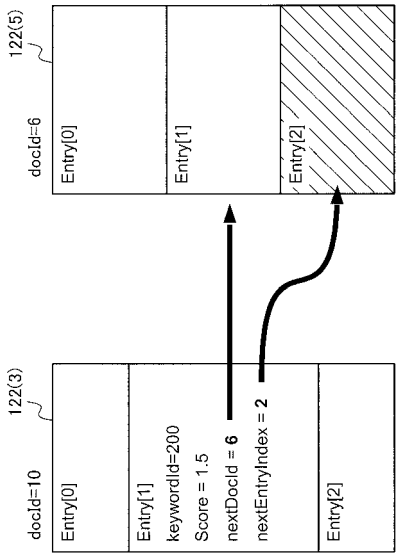
【図６】


開始
S501
インデックスdi を作成
S502
インデックスdi をDtoKリストの先頭に追加
S503
未処理の単語がある ？
No
Yes
終了
S504
未処理の単語を一つ選択
S505
選択した単語が
単語リスト中にある ？
Yes
No
S506
単語リストに選択した単語のエントリを追加
S507
単語リストにおいて、
選択した単語のエントリの
リンク先の情報を更新
S508
インデックスdi において、
選択した単語のエントリの
リンク先の情報を更新

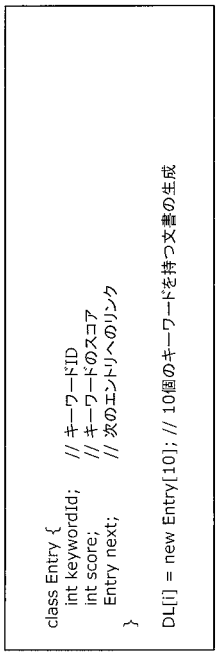
【図７】


class Entry {
int keywordId; // キーワードID
int score; // キーワードのスコア
int nextDocId; // 次のリンク先文書ID
int nextEntryIndex; // 次のリンク先文書内でのエントリの位置
}
DL[i] = new Entry[10]; // 10個のキーワードを持つ文書の生成
(a)
java.util.TreeMap<Integer, Entry[]> DLList = new java.util.TreeMap<>();
(b)
int newDocId = 100;
Entry[] newDL = new Entry[10];
DLList.put(newDocId, newDL);
(c)
DLList.pollFirstEntry(); // 最小の文書IDを削除
(d)

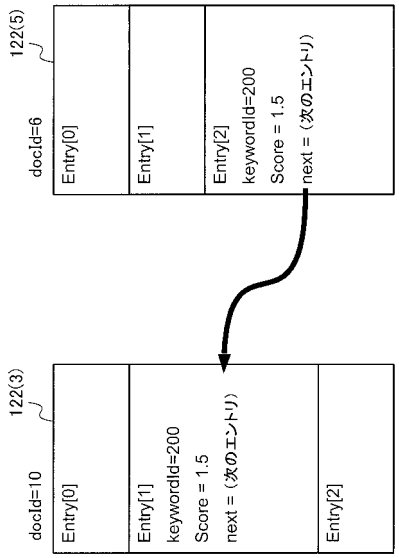
【図８】


122(5)
docId=6
Entry[0]
Entry[1]
Entry[2]
122(3)
docId=10
Entry[0]
Entry[1]
keywordId=200
Score = 1.5
nextDocId = 6
nextEntryIndex = 2
Entry[2]

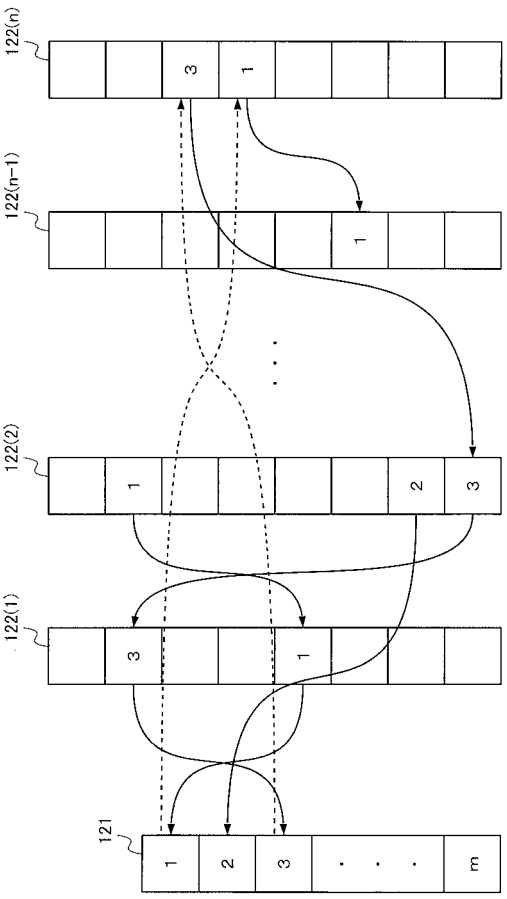
【図９】



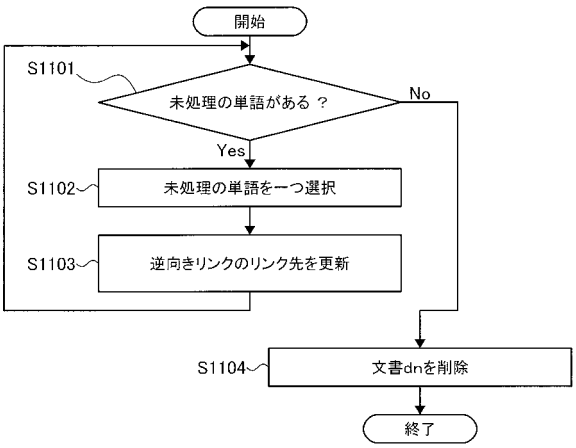
【図１０】



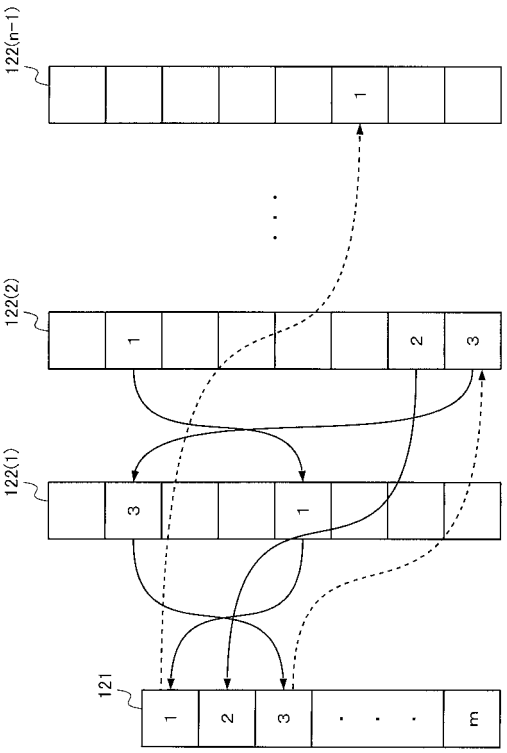
【図１１】



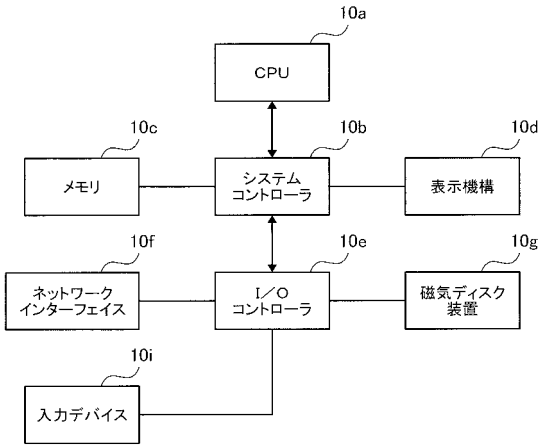
【図１２】



【図１３】


122(n-1)
1
122(2)
1
2
3
122(1)
3
1
121
1
2
3
m

【図１４】


10a
CPU
10c
メモリ
10b
システム
コントローラ
10d
表示機構
10f
ネットワーク
インターフェイス
10e
I／O
コントローラ
10g
磁気ディスク
装置
10i
入力デバイス

フロントページの続き

(74)代理人　100112690
弁理士　太佐　種一
(74)復代理人　100104880
弁理士　古部　次郎
(74)復代理人　100118108
弁理士　久保　洋之
(72)発明者　吉田　一星
東京都江東区豊洲五丁目６番５２号　ＮＢＦ豊洲キャナルフロント　日本アイ・ビー・エム株式会社　東京基礎研究所内
(72)発明者　榎　美紀
東京都江東区豊洲五丁目６番５２号　ＮＢＦ豊洲キャナルフロント　日本アイ・ビー・エム株式会社　東京基礎研究所内