(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2011-159100
(P2011-159100A)
(43)公開日　平成23年8月18日(2011.8.18)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 17/30	(2006.01)	G06F	17/30	350C	5B075
		G06F	17/30	170A	

審査請求 未請求　請求項の数 10　ＯＬ　（全 17 頁）

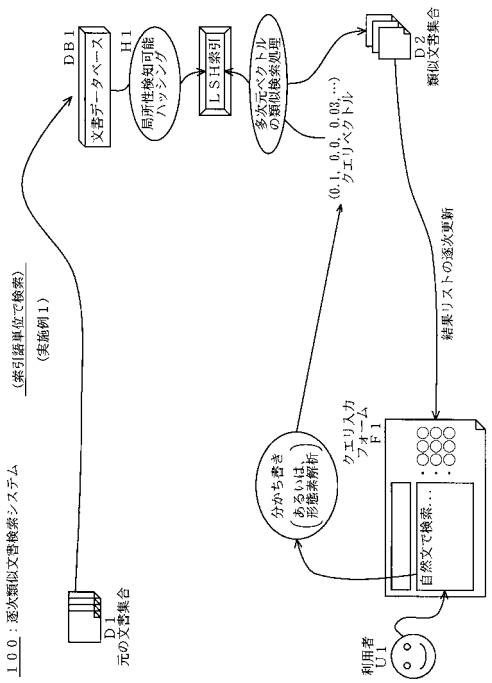
(21)出願番号　特願2010-20137(P2010-20137)
(22)出願日　平成22年2月1日(2010.2.1)

(71)出願人　000004226
日本電信電話株式会社
東京都千代田区大手町二丁目３番１号
(74)代理人　100087446
弁理士　川久保　新一
(72)発明者　江田　毅晴
東京都千代田区大手町二丁目３番１号　日本電信電話株式会社内
(72)発明者　別所　克人
東京都千代田区大手町二丁目３番１号　日本電信電話株式会社内
(72)発明者　内山　俊郎
東京都千代田区大手町二丁目３番１号　日本電信電話株式会社内

最終頁に続く

(54)【発明の名称】逐次類似文書検索装置、逐次類似文書検索方法およびプログラム

(57)【要約】
【課題】ニュースやブログ記事、ソーシャルネットワークの日記等、日々大量に生産されているテキストのうちで、所定の文書が類似している文書を検索する場合、利便性が高い逐次類似文書検索装置、逐次類似文書検索方法及びプログラムを提供することを目的とする。
【解決手段】類似文書を逐次的に検索する逐次的類似文書検索手段と、上記逐次的類似文書検索手段が検索した検索結果を更新する更新手段とを有することを特徴とする逐次類似文書検索装置である。
【選択図】図３



【特許請求の範囲】
【請求項１】
類似文書を逐次的に検索する逐次的類似文書検索手段と；
上記逐次的類似文書検索手段が検索した検索結果を更新する更新手段と；
を有することを特徴とする逐次類似文書検索装置。
【請求項２】
請求項１において、
検索文の索引語境界を検出する索引語境界検出手段を有し、
上記逐次的類似文書検索手段は、上記索引語境界検出手段が上記検索文の索引語境界を検出する度に、索引語単位で逐次的に類似文書を検索する手段であることを特徴とする逐次類似文書検索装置。
【請求項３】
請求項２において、
上記索引語境界検出手段は、局所性検知可能ハッシングを利用して、検索文の索引語境界を検出する手段であることを特徴とする逐次類似文書検索装置。
【請求項４】
請求項２において、
上記索引語境界検出手段は、概念ベース法による類似文書検索を実現する手段であることを特徴とする逐次類似文書検索装置。
【請求項５】
請求項４において、
検索文の索引語境界を検出する直前における上記検索文の重心ベクトルと上記検索文における単語ベクトルとを記憶する記憶手段を有し、
上記索引語境界検出手段は、上記直前の状態から新規追加、削除された索引語のみを検索する手段であり、
上記更新手段は、上記重心ベクトルを更新する手段であることを特徴とする逐次類似文書検索装置。
【請求項６】
逐次的類似文書検索手段が、類似文書を逐次的に検索し、記憶手段に記憶する逐次的類似文書検索段階と；
上記逐次的類似文書検索段階で検索された検索結果を更新する更新段階と；
を有することを特徴とする逐次類似文書検索方法。
【請求項７】
請求項６において、
検索語境界検索手段が、検索文の索引語境界を検出し、記憶手段に記憶する検索語境界検索段階を有し、
上記逐次的類似文書検索段階は、上記索引語境界検出段階で上記検索文の索引語境界が検出される度に、索引語単位で逐次的に類似文書を検索する段階であることを特徴とする逐次類似文書検索方法。
【請求項８】
請求項７において、
上記索引語境界検出段階は、局所性検知可能ハッシングを利用して、検索文の索引語境界を検出する段階であることを特徴とする逐次類似文書検索方法。
【請求項９】
請求項７において、
上記索引語境界検出段階は、概念ベース法による類似文書検索を実現する段階であることを特徴とする逐次類似文書検索方法。
【請求項１０】
請求項６～請求項９のいずれか１項に記載の逐次類似文書検索方法をコンピュータに実行させるプログラム。

【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、逐次類似文書検索装置、逐次類似文書検索方法およびプログラムに関する。
【背景技術】
【０００２】
ウェブの発展によって、ニュースやブログ記事、ソーシャルネットワークの日記等のテキストが、日々大量に生産されている。記録されたテキストは、いずれかの人が読むためのものであるから、誰かによって既に書かれたテキストと類似しているかどうかは、そのテキストの価値に係る根源的な問題である。
【０００３】
故に、計算機の発展によって可能になった大量の文書からなる文書データベースにおける類似文書検索技術は、非常に重要な技術であると言える。実際、その応用範囲も幅広く、特許文書や論文等に対しても適用が試みられている。
【０００４】
文書間の類似度を測る場合、その文書で利用されている単語を、シンボルの集合としてベクトル表現し、距離を測る方法があり、類似文書検索に広く用いられている。しかし、単語すなわち言葉は人が作り出したものであり、その意味の多様性をシンボルとして扱うだけでは不十分である。
【０００５】
そこで、眼前の文書には現れない、その単語が持つ豊かな意味あいを考慮した文書の表現方法として、概念ベース法が提案されている（たとえば、非特許文献１参照）。概念ベース法は、大量のトレーニングデータを用いて、単語をベクトルとして表現する（単語概念ベクトルで表現する）。文書は、登場する単語概念ベクトルの重心として表現することができ、つまり、ある単語を単語概念ベクトルで表現すると、その単語の使われ方を表現している。上記単語概念ベクトルによって、人間が作り出した言葉の豊かな意味合いを考慮した類似文書検索が可能である。
【０００６】
概念ベース法によって、類似文書検索は、数百から数千次元の密なベクトル空間内での類似検索に置き換えられる（ベクトルの次元はアプリケーション依存で決定される）。しかし、従来、高次元ベクトル空間内での類似検索は、「次元の呪い」によって、高速に処理することが困難である。概念ベース法においても、類似文書を検索する場合、データベース内の全文書との突き合わせの後にランキングを行うというナイーブな手法で実現されるので、数百万件の文書データベースに対して、数秒の検索時間を要する。
【０００７】
一方、文書処理技術とは別の分野であるデータベース分野から、「次元の呪い」を克服する近似的なベクトルの類似検索技術の研究が、１９９８年頃から始まっている（たとえば、非特許文献６参照）。提案されている様々な技術の中では、局所性検知可能ハッシュ（ＬＳＨ）（たとえば、非特許文献４参照）が最も有望である。局所性検知可能ハッシュは、多次元ベクトルを近似して複数のハッシュを用いて索引付けを行う。これによって、検索精度を確率的に保証しながら、検索の計算コストは、（ハッシュ個数）×（サンプリングビット数）で済む。
【０００８】
理論上、次元数に大きく依存せずに高速に類似検索が可能であるので、ウェブにおける情報推薦や画像検索等への応用が期待されている。
【０００９】
局所性検知可能ハッシュ（ＬＳＨ）は、近似アルゴリズムであるものの、確率的に精度が保証され（たとえば、非特許文献３、４参照）、非常に高速に近傍（類似）検索を実現することができる。 理論としてのフレームワークは、非特許文献３で整理され、非特許文献４で、Ｌ１ノルムが定義された多次元ベクトル空間をハミング空間に写像する局所性

検知可能ハッシュ（ＬＳＨ）が実装されている。また、最近は、Ｌ１ノルムだけでなく、安定分布を利用するＬ２ノルム版や、コサイン類似度版、Ｊａｃｃａｒｄ係数版も提案されている。

【００１０】

ところで、従来技術において、キーワードによる逐次検索（インクリメンタルサーチ）は、英語・日本語を問わず、広く普及し、利用されている（各種検索エンジンや、非特許文献２参照）。

【００１１】

「逐次検索」は、検索したい単語をすべて入力して検索するのではなく、ユーザが文字を１文字入力する度に、即座に候補を表示させる検索方法である。上記逐次検索は、ユーザの入力に従って動的に検索が進行する「Ｄｙｎａｍｉｃ　Ｑｕｅｒｙ」の一種であり、検索の効率化だけではなく、テキスト編集の効率化にもつながる（たとえば、非特許文献２参照）。通常の逐次検索は、検索クエリの部分文字列による部分一致検索であり、入力文字列をシンボルとみなし、検索対象もシンボルとみなした上で、一致箇所に早く辿り着くための技術である。

【００１２】

キーワードをシンボルとみなした通常の一致検索に対して、言語の持つ豊かな概念を利用した類似文書検索技術が提案されている（たとえば、非特許文献１参照）が、しかし、非特許文献１に記載されている発明では、高次元概念ベクトル同士の距離計算のコストを削減することができない。

【先行技術文献】

【非特許文献】

【００１３】

【非特許文献１】別所克人、内山俊郎、片岡良治著「単語・意味属性間共起に基づく概念ベースの拡張方式」、情報処理学会研究報告　2006-ICS-144 2006 /7/28、pp.29－34.

【非特許文献２】高林 哲、小松 弘幸、増井 俊之著「Ｍｉｇｅｍｏ：日本語のインクリメンタル検索」、情報処理学会論文誌、Vol. 43、No. 12、pp.3698－3705、December, 2002.

【非特許文献３】Kevin S. Beyer, Jonathan Goldstein, Raghu Ramakrishnan, Uri Shaft著「When Is“Nearest Neighbor”Meaningful?」ICDT 2005

【非特許文献４】Piotr Indyk, Rajeev Motwani著“Approximate nearest neighbors: towards removing the curse of dimensionality”Annual ACM Symposium on Theory of Computing 1998

【非特許文献５】Aristides Gionis, Piotr Indyk, Rajeev Motwani著“Similarity Search in High Dimensions via Hashing” Very Large Data Bases 1999

【非特許文献６】Roger Weber, Hans-Jorg Schek, Stephen Blott著“A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces” Very Large Data Bases 1998

【発明の概要】

【発明が解決しようとする課題】

【００１４】

上記従来例では、単語の概念を用いた類似文書検索において、「次元の呪い」に起因する検索速度低下が引き起こす類似文書検索インタフェースのレスポンスの悪さが生じ、つまり、類似文書検索の利便性が低いという問題がある。

【００１５】

また、上記従来例では、ユーザが質問文の作成を完成した後に検索を開始するので、ユーザによる質問文の作成の途中で類似文書検索の結果を得ることができず、類似文書検索の結果を質問文完成まで待つ必要があり、この意味でも、類似文書検索の利便性が低いという問題がある。

【００１６】

さらに、上記従来例では、検索精度が低下し、検索サーバのＣＰＵコストが下がらず、高速な逐次類似文書検索ができないという問題がある。

【００１７】

本発明は、類似文書検索において利便性が高い逐次類似文書検索装置、逐次類似文書検索方法及びプログラムを提供することを目的とする。

【課題を解決するための手段】

【００１８】

本発明は、類似文書を逐次的に検索する逐次的類似文書検索手段と、上記逐次的類似文書検索手段が検索した検索結果を更新する更新手段とを有することを特徴とする逐次類似文書検索装置である。

【発明の効果】

【００１９】

本発明によれば、類似文書検索において利便性が高いという効果を奏する。

【図面の簡単な説明】

【００２０】

【図１】本発明の実施例１である逐次類似文書検索システム１００の概略を示すブロック図である。

【図２】逐次類似文書検索システム１００におけるクライアントＣＬ１と、類似文書検索アプリケーション１０とのハードウェア構成を示す図である。

【図３】逐次類似文書検索システム１００の動作の説明図であり、ＬＳＨ索引を用いた逐次類似文書検索処理の処理概要を示す図である。

【図４】実施例１の動作を示すフローチャートである。

【図５】本発明の実施例２である逐次類似文書検索システム２００の動作の説明図であり、ＬＳＨ索引を用いた逐次類似文書検索において、概念ベースを用いた処理の説明図である。

【図６】実施例２の動作を示すフローチャートである。

【図７】本発明の実施例３である逐次類似文書検索システム３００において、ＬＳＨ索引を用いた逐次類似文書検索において、概念ベースを用い、単語概念ベクトルのキャッシュを利用した処理の概要を示す図である。

【図８】実施例３の動作を示すフローチャートである。

【発明を実施するための形態】

【００２１】

発明を実施するための形態は、以下の実施例である。

【実施例１】

【００２２】

図１は、本発明の実施例１である逐次類似文書検索システム１００の概略を示すブロック図である。

【００２３】

逐次類似文書検索システム１００は、文字単位ではなく、索引語単位で検索する実施例である。

【００２４】

なお、上記逐次類似文書検索は、類似文書検索を、インクリメンタルに行うことであり、入力検索文ｑの索引語境界を認識し、索引語が追加されたタイミングで、動的に検索結果の取得更新を行う。

【００２５】

逐次類似文書検索システム１００は、類似文書検索アプリケーション１０と、ネットワークＮＷ１と、クライアントＣＬ１、ＣＬ２、ＣＬ３とによって構成され、クライアントＣＬ１、ＣＬ２、ＣＬ３のそれぞれには、利用者Ｕ１、Ｕ２、Ｕ３が対応している。

【００２６】

類似文書検索アプリケーション１０は、類似文書検索装置の例であり、ルータ１１と、

ＬＡＮ１２と、アプリケーションサーバ１３と、データベースサーバ１４、１５と、類似文書検索エンジン１６とを有する。類似文書検索エンジン１６は、ＬＳＨ構築エンジン１６１と、問合せ処理エンジン１６２とを有する。

【００２７】

図２は、逐次類似文書検索システム１００におけるクライアントＣＬ１と、類似文書検索アプリケーション１０とのハードウェア構成を示す図である。

【００２８】

クライアントＣＬ１は、通信手段３１と、記憶手段３２と、データ処理手段３３と、ユーザインタフェース３４とを有する。

【００２９】

類似文書検索アプリケーション１０は、通信インタフェース１７と、制御手段１８と、記憶手段１９とを有し、入力装置２１と出力装置２２とに接続されている。

【００３０】

制御手段１８は、ＬＳＨ構築手段１８１と、問合せ処理手段１８２とを有する。

【００３１】

記憶手段１９は、ＲＯＭ１９１と、ＲＡＭ１９２と、ＨＤＤ１９３と、ＳＳＤ１９４とを有する。

【００３２】

図３は、逐次類似文書検索システム１００の動作の説明図であり、ＬＳＨ索引を用いた逐次類似文書検索処理の処理概要を示す図である。

【００３３】

なお、ＬＳＨは、局所性検知可能ハッシュ（Locality Sensitive Hashing）であり、近似近傍検索を、ハッシュを用いて実現する方法であり、ハミング距離、ユークリッド距離、Ｌ２ノルム、コサイン類似度に対応したハッシュ構成方法が提案されている。

【００３４】

元の文書集合Ｄ１は、文書ベクトルモデルなどを利用して文書データベース化する。文書データベースＤＢ１に対して、局所性検知可能ハッシングＨ１を適用し、ＬＳＨ索引を構築する。以上は、前処理として実行する。

【００３５】

利用者Ｕ１が検索文ｑを入力し、リアルタイムに分かち書きを行い、同様に文書ベクトルモデルなどを利用して、検索文ｑを多次元ベクトルに変換する。なお、上記多次元ベクトルは、物体の位置や形状、画像、動画、テキスト等の特徴を、ユークリッド空間のベクトルと見做して表現したものであり、その次元数は、計測機器やアプリケーションに依存して決められる。たとえば、概念ベースの単語概念ベクトルは、多次元ベクトルである。

【００３６】

ＬＳＨ索引を用いて近似類似検索を実行し、類似文書集合Ｄ２を結果として得る。最後に、結果リストを逐次更新しながら、利用者Ｕ１に表示する。

【００３７】

逐次類似文書検索システム１００は、局所性検知可能ハッシングＨ１を用いて高速な類似文書検索を実現する。さらに、高速な検索処理を利用して、従来、不可能であった逐次類似文書検索（インクリメンタル類似文書検索）を実現し、類似文書検索のインタフェースを改善する。

【００３８】

また、一般に利用者Ｕ１は、検索文ｑを逐次的に追加するので、検索文ｑに既に含まれている単語ベクトルを、キャッシュすることができ、索引語データベースＤＢ２へ問い合わせる場合、事前の分かち書き結果に追加された単語のみを問い合わせればよい。すなわち、索引語データベースＤＢ２への問合せも高速に実現可能である。以上によって、類似文書検索が高速に処理されるので、逐次類似文書検索を実現することができる。

【００３９】

上記実施例によれば、長い自然文を検索文ｑとする検索が高速に実現可能であり、これ

によって、フリーフォームによる文書編集環境での逐次類似文書検索が可能である。たとえば、「教えてｇｏｏ」に代表されるＱ＆Ａサイトにおいて、質問者が新しい質問文を作っている最中でも、質問文を追記するにつれ、逐次的検索することによって、似た質問文を即座に提示することができる。つまり、検索文を作成中に、似た質問が利用者に提示されるので、質問しなくても疑問点が解決されることがあり、この場合には、その似た質問を繰り返して実行することを回避することができ、質問者・サイト運営者の双方にとってメリットである。

【００４０】

また、提案技術の適用領域は、日本語に限定されない。英語のような区切りの明確な言語の場合、分かち書きせずに、クエリベクトルを求めることができるので、より高速に逐次類似文書検索を実現することができる。

【００４１】

類似文書検索システム１００は、ニュースやウェブログ等の様々なコンテンツの類似文書検索エンジン１６として利用することができる。

【００４２】

図１に示す逐次類似文書検索システム１００は、類似文書検索アプリケーション１０内部で利用した例である。

【００４３】

類似文書検索エンジン１６は、アプリケーションサーバ１３から利用され、ＬＳＨ構築エンジン１６１と、問合せ処理エンジン１６２とによって構成されている。これらの処理部は、単一のサーバ内で実現することができるだけでなく、複数台で分散構成することができる。

【００４４】

次に、実施例１の動作について説明する。

【００４５】

［逐次検索処理］

図４は、実施例１の動作を示すフローチャートである。

【００４６】

Ｓ１で、文書データベースＤＢ１に対して構築したＬＳＨ索引ｉｄｘと、ある時点での検索文ｑとを入力する。

【００４７】

図３に示す文書データベースＤＢ１は、概念ベース法によるものに限定する必要はない。たとえば、単語をシンボルとみなし、その頻度で重みをつけるｔｆ＊ｉｄｆ法等を利用するようにしてもよい。このようにすることによって、それぞれの文書を多次元ベクトルとすることができるので、局所性検知可能ハッシングＨ１を用いて索引付けすることができる。

【００４８】

次に、Ｓ２で、検索文ｑを分かち書き処理し、結果単語列Ｖｑを生成する。たとえば、単語をシンボルとみなし、その頻度で重みをつけるｔｆ＊ｉｄｆ法を利用することが考えられる。そして、Ｓ３で、検索文ｑの末尾が索引語境界であるかどうかを判定し、索引語境界でないと判定すれば、Ｓ４で、利用者Ｕ１が検索分ｑ（クエリ）を追記するのを待つ。索引語境界であると判定すれば、Ｓ５で、結果単語列Ｖｑを用い、予め構築されているＬＳＨ索引ｉｄｘを利用し、類似検索処理を行う。

【００４９】

上記類似検索は、大量の多次元ベクトルが格納された多次元データベースから，与えられた検索文ｑに近い（似た）ベクトルを取得する検索である。近さは、距離（類似度）によって決められる。特に、文書が検索対象のベクトルである場合、文書が検索対象のベクトルであれば、類似文書検索と呼ぶ。上記距離は、ある２つの多次元ベクトルの間に定義される尺度のうちで、距離の公理を満たすものである。

【００５０】

検索文ｑの末尾が索引語境界であるかどうかを判定する場合、後述の［例１］に示すように、形態素解析の結果である単語の品詞が、名詞、サ変動詞であれば、索引語境界であると判定するか、または、後述の［例２］に示すように、索引語データベースＤＢ２に単語が存在すれば、当該単語が検索後境界であると判定する等によって、検索文ｑの末尾が索引語境界であるかどうかを判定することができる。

【００５１】

「検索文ｑの末尾が単語境界であるかどうかを判定する例」

［例１］：形態素解析し、「名詞」と「サ変動詞」とを用いることによって、文書集合を索引付けする（索引語境界であると判断する）方法

上記［例１］では、検索文ｑを形態素解析し、形態素解析した末尾が、名詞またはサ変動詞であれば、検索語境界であると判断する。たとえば、検索文ｑとして「今日の天気は、いまい」を形態素解析すると、「今日：名詞、の：格助詞、天気：名詞、は：連用助詞、い、：動詞語幹、まい：動詞接尾辞」である。この検索文ｑの末尾「まい」は、動詞接尾辞であり、名詞、サ変動詞のいずれでもないので、上記検索文ｑの末尾は、索引語境界ではないと判断する。

【００５２】

形態素解析の結果の単語列の最後の単語の品詞が、索引付け（索引語境界であると判断する）対象の品詞（名詞またはサ変動詞）であれば、上記「形態素解析の結果の単語列の最後の単語」が、索引語境界である。下記のように文字が逐次的に入力された場合、３つの索引語境界ｘ、ｙ、ｘが出現し、索引語境界でのみ、類似文書を検索するので、類似文書検索を３回実行すれば足り、検索サーバ（図５に示す類似文書検索アプリケーション１０の類似文書検索エンジン１６）のＣＰＵコストが削減される。

・今日の天（名詞）…索引語境界ｘ

つまり、「今日の」に引き続いて、「天」を逐次的に入力した場合、入力された「天」は名詞であり、名詞は、索引付け対象の品詞であるので、入力された「天」は、索引語境界であり、この境界を索引語境界ｘと表現する。

・今日の天気（名詞）…索引語境界ｙ

つまり、「今日の天」に引き続いて、「気」を逐次的に入力した場合、入力された「気」は名詞であり、名詞は、索引付け対象の品詞であるので、入力された「気」は、索引語境界であり、この境界を索引語境界ｙと表現する。

・今日の天気は（連用助詞）

つまり、「今日の天気」に引き続いて、「は」を逐次的に入力した場合、入力された「は」は連用動詞であり、連用動詞名詞は、索引付け対象の品詞ではないので、入力された「は」は、索引語境界ではない。

・今日の天気は、い（動詞語幹）

・今日の天気は、いま（名詞）…索引語境界ｚ

つまり、「今日の天気は、」に引き続いて、「いま」を逐次的に入力した場合、入力された「いま」は名詞であり、名詞は、索引付け対象の品詞であるので、入力された「いま」は、索引語境界であり、この境界を索引語境界ｚと表現する。

・今日の天気は、いまい（動詞接尾辞）

・今日の天気は、いまいち（連用詞）

・今日の天気は、いまいちだ（判定詞）

・今日の天気は、いまいちだっ（終助詞）

・今日の天気は、いまいちだった（判定詞）

［例２］：索引語データベースＤＢ２を用いて索引語境界であるかどうかを判断する方法

上記［例２］は、分かち書きのみの結果を用いて、分かち書き結果の最後の単語が、索引語データベースＤＢ２に含まれていれば、索引語境界であると判断する方法である。［例２］では、形態素解析まで実行する必要はない。なお、索引語データベースＤＢ２は、ＨＤＤ１９３等に格納されている。

・今日の「天」（含まれる）索引語境界
つまり、「今日の」に引き続いて、「天」を逐次的に入力した場合、入力された最後の単語「天」が、索引語データベースＤＢ２に含まれているので、入力された「天」は、索引語境界であり、この境界を索引語境界ｘと表現する。
・今日の「天気」（含まれる）索引語境界
つまり、「今日の天」に引き続いて、「気」を逐次的に入力した場合、入力された最後の単語「天気」が、索引語データベースＤＢ２に含まれているので、入力された「気」は、索引語境界である。
・今日の天気「は」（含まれない）
つまり、「今日の天気」に引き続いて、「は」を逐次的に入力した場合、入力された最後の単語「は」が、索引語データベースＤＢ２に含まれていないので、入力された「は」は、索引語境界ではない。
・今日の天気は、「い」（含まれない）
・今日の天気は、「いま」（含まれない）
「いま」が、漢字の「今」であれば、「今」が索引語データベースＤＢ２に含まれているので、入力された「今」は、索引語境界である。しかし、「いま」は、索引語データベースＤＢ２に含まれていないので、入力された「いま」は、索引語境界ではない。
・今日の天気は、い「まい」（含まれない）
・今日の天気は、「いまいち」（含まれる）索引語境界
・今日の天気は、いまいち「だ」（含まれない）
・今日の天気は、「いまいちだっ」（含まれない）
・今日の天気は、いまいち「だった」（含まれない）
なお、上記「（含まれる）」は、分かち書き結果の最後の単語が、索引語データベースＤＢ２に含まれていることを意味する。また、上記「（含まれない）」は、分かち書き結果の最後の単語が、索引語データベースＤＢ２に含まれていないことを意味する。

【００５３】
索引語境界以外の分かち書き結果に含まれている索引語集合と、その直前の索引語境界での分かち書き結果に含まれている索引語集合とが同一であることに注意を要する。換言すれば、検索語境界の状態で類似文書検索しても、検索語境界以外の状態で類似文書検索しても、検索結果が変化しない。つまり、検索語境界の状態でのみ類似文書検索しても、検索精度が低下しない。すなわち、検索語境界以外の状態で類似文書検索せずに、これによって、検索回数を減らしても、検索結果に影響がなく、したがって、検索精度が低下しない。なお、文字を入力する度に、索引語境界の検出処理を実行し、上記直前は、検索文の索引語境界を検出する直前であり、今実行した境界検出の直前の境界検出である。

【００５４】
次に、Ｓ６で、類似検索の結果ベクトル集合Ｒｑを、利用者Ｕ１が見ているインタフェースに逐次的に追加する。これによって、逐次検索を実現することができる。

【００５５】
たとえば、上記［例１］において、索引語境界ｘ、ｙ、ｚまでのそれぞれの検索文ｑ（クエリ）で検索すると、下記の検索結果を得ることができる。

【００５６】
索引語境界ｘ…今日の天→今日、天　［検索結果］：「今日は天気が良いね」、「今日、天に召されました」
つまり、索引語境界ｘまでの検索文ｑである「今日の天」で検索すると、この検索結果（ランキングの上位２つの検索結果）は、「今日は天気が良いね」、「今日、天に召されました」であったとする。

【００５７】
索引語境界ｙ…今日の天気→今日、天気　　［検索結果］：「今日の天気はいかが？」、「今日は天気が良いね」
つまり、索引語境界ｘまでの検索文ｑである「今日の天気」で検索すると、この検索結

果（ランキングの上位２つの検索結果）は、「今日の天気はいかが？」、「今日は天気が良いね」であったとする。

【００５８】

索引語境界ｚ…今日の天気は、いま→今日、天気、いま　［検索結果］：「今日の天気は、いまからどうなる？」、「今日の天気はいかが？」

検索文が追記されればされる程、より多くのキーワードの特徴を利用できるので、検索結果の精度が高くなることが期待できる。

【００５９】

最後に、Ｓ７で、検索文ｑに更新があると判定すれば、すなわち利用者Ｕ１が検索文ｑ（クエリ）を修正した場合、クエリからベクトルを生成するステップＳ２に戻り、同じ処理を繰り返す。

【００６０】

実施例１は、多次元ベクトル空間をハミング空間に写像する局所性検知可能ハッシングＨ１を用いて、性能の課題を解決しつつ、逐次類似文書検索を実現する発明であり、逐次類似文書検索では、入力検索文ｑの索引語境界を認識し、索引語が追加されたタイミングで、検索結果の取得更新を動的に行い、検索文ｑを入力するにつれ、逐次的に検索結果を得ることができる。

【００６１】

実施例１によれば、検索精度が低下せず、検索サーバのＣＰＵコストを下げて、高速な逐次類似文書検索が可能である。

【００６２】

なお、実施例１において、索引語単位で逐次的に類似文書を検索しなくてもよく、また、検索文の索引語境界を検出しなくてもよい。つまり、類似文書を逐次的に検索すれば足り、上記実施例は、類似文書を逐次的に検索する逐次的類似文書検索手段と、上記逐次的類似文書検索手段が検索した検索結果を更新する更新手段とを有する逐次類似文書検索装置の例である。このようにすれば、ユーザによる質問文の作成の途中で類似文書検索の結果を得ることができ、類似文書検索の結果を質問文完成まで待つ必要がなく、類似文書検索の利便性が高い。

【実施例２】

【００６３】

本発明の実施例２である逐次類似文書検索システム２００は、索引語単位で検索し、しかも概念ベース法を用いる実施例である。

【００６４】

［概念ベースを用いた逐次検索処理］

逐次類似文書検索システム２００のハードウェアは、図１、図２に示す逐次類似文書検索システム１００と同様である。

【００６５】

次に、本発明の実施例２の動作について説明する。

【００６６】

図５は、本発明の実施例２である逐次類似文書検索システム２００の動作の説明図であり、ＬＳＨ索引を用いた逐次類似文書検索において、概念ベースを用いた処理の説明図である。

【００６７】

上記概念ベースは、単語と意味属性共起行列とに、特異値分解を施すことによって抽出されたデータベースである。意味属性付与機能を持つ形態素解析器を利用することによって、膨大なトレーニングデータの意味を反映させた単語の意味ベクトルを構築することができる。登場する単語ベクトルの重心として、文書を表現することができ、文書の意味的類似性に基づいた概念検索を、ベクトル間の距離の近さを用いて実現できる。

【００６８】

元の文書集合Ｄ１から、概念ベース法を用いて文書データベースＤＢ１と索引語データ

ベースＤＢ２とを構築する。文書データベースＤＢ１に対して、局所性検知可能ハッシングＨ１を適用し、ＬＳＨ索引を構築する。以上は、前処理として実行する。

【００６９】

利用者Ｕ１が検索文ｑ（クエリ）を入力し、リアルタイムに分かち書きを行い、索引語データベースＤＢ２を利用し、入力された検索文ｑ（クエリ）を図５に示すクエリ概念ベクトルに変換する。ＬＳＨ索引を用いて近似類似検索を実行し、類似文書集合Ｄ２を結果として得る。最後に、利用者Ｕ１に結果リストを逐次更新を行いながら表示する。

【００７０】

図６は、実施例２の動作を示すフローチャートである。

【００７１】

実施例２では、Ｓ１１で、概念ベース法によって構築した単語概念ベース（図５参照、単語概念ベクトルを単語名で検索できるようにしたデータベース）を用意する（非特許文献１参照）。Ｓ２で、概念ベース法の類似文書検索処理と同様に、まず、検索文ｑを分かち書き出し、単語集合Ｖｑを得る。次に、単語集合Ｖｑ中のそれぞれの単語の概念ベクトルを、単語概念ベースから取得する。

【００７２】

そして、Ｓ１２で、これらの単語概念ベクトル集合の重心を求める。重心は、たとえば、単語概念ベクトルの集合Ｄｗを、Ｄｗ＝｛ｖ１，ｖ２，…ｖｍ｝としたときに、重心ベクトルG_qを、

【００７３】

【数１】

$$G_q = \frac{1}{m}\sum v_i \qquad \cdots式（1）$$

と表すことができる。なお、ｍは、単語概念ベクトルの数である。Ｓ１３で、この重心ベクトルG_qを検索キーとして、ＬＳＨ索引ｉｄｘに対して検索処理を行い、Ｓ１４で、検索結果を逐次的にインタフェースに追加する。さらに、検索文ｑに更新があれば、Ｓ２に戻り、再度検索を行い、結果を逐次的に更新する。

【００７４】

実施例２によれば、索引語境界を検出する場合、概念ベース法による類似文書検索を実現するので、語の意味合いを考慮した逐次類似文書検索が可能である。つまり、実施例２は、概念ベース法によってベクトル化した文書データベースンＤＢ１に対して、予め、局所性検知可能ハッシングＨ１を用いて、索引付けする（索引語境界を検出する）。また、概念ベース法によって構築した単語概念ベクトル集合も、単語名に索引（ハッシュやＢ木等）を付与して索引語データベースＤＢ２として索引付けする。利用者Ｕ１が検索文ｑの文字列を入力する度に、単語集合を取得する。語の切れ目の無い日本語等の場合、即座に検索文ｑ（検索クエリ）の分かち書きを行い、単語集合を取得する。直前状態から単語の追加、削除があれば、分割されたそれぞれの単語を用いて、索引語データベースＤＢ２を検索し、このデータベースに含まれている単語集合の重心ベクトルを求め、これを検索クエリベクトルとする。なお、この検索クエリベクトル（＝クエリベクトル）は、クエリ概念ベクトルの上位概念であり、検索クエリベクトルのうちで、単語概念ベースを用いて作成したベクトルが、クエリ概念ベクトルである。

【００７５】

上記検索クエリベクトルを検索キーにして、文書データベースンＤＢ１に対して、問合せする。文書データベースンＤＢ１は、局所性検知可能ハッシングＨ１によって、高速に類似文書集合Ｄ２を返却可能であり、擬似的に類似文書集合Ｄ２を逐次的に利用者Ｕ１に返し、見せることができる。通常の逐次検索（インクリメンタルサーチ）とは異なり、検

索結果の更新を文字単位ではなく、単語単位で検索結果を更新することによって、サーバの負荷を下げることができる。

【実施例３】

【００７６】

［単語ベクトルのキャッシュを用いた逐次検索処理］

本発明の実施例３である逐次類似文書検索システム３００は、索引語単位で検索し、概念ベース法を用い、しかも、直前の重心ベクトルと単語とを記憶し、差分のみによって、次の重心ベクトルを更新する実施例である。

【００７７】

次に、逐次類似文書検索システム３００の動作について説明する。

【００７８】

図７は、本発明の実施例３である逐次類似文書検索システム３００において、ＬＳＨ索引を用いた逐次類似文書検索において、概念ベースを用い、単語概念ベクトルのキャッシュを利用した処理の概要を示す図である。

【００７９】

元の文書集合Ｄ１から、概念ベース法を用いて文書データベースＤＢ１と索引語データベースＤＢ２とを構築する。文書データベースＤＢ１に対して、局所性検知可能ハッシングＨ１を適用し、ＬＳＨ索引を構築する。以上は、前処理として実行する。

【００８０】

利用者Ｕ１が検索文ｑを入力し、リアルタイムに分かち書きを行い、検索文ｑ中に、直前の検索文ｑからの差分があるかどうかを検出する。新規追加または削除があった単語のみを、索引語データベースＤＢ２を用いて問い合わせ、直前の検索文ｑと合わせて用いて、クエリ概念ベクトルを構築する。検索文ｑの集合は、次回の問い合わせに備えて保存する。ＬＳＨ索引を用いて近似類似検索を実行し、類似文書集合Ｄ２を結果として得る。最後に、利用者Ｕ１に結果リストを逐次更新を行いながら表示する。

【００８１】

図８は、実施例３の動作を示すフローチャートである。

【００８２】

通常の検索時には、利用者Ｕ１は検索文ｑを追記していくことが多い。また、本技術が想定する、日記記入テキストエリアやＱ＆Ａサイトにおける質問文作成テキストエリア等、フリーフォームのテキスト編集環境では、ある程度の長い文書を追記する。そこで、逐次類似文書検索する直前の重心ベクトルと単語を記憶し、Ｓ２１で、更新前の分かち書き結果から取得した索引語集合と、更新後の分かち書き結果から取得した索引語集合との差分を検出し、この検出された差分のみについて検索することによって、単語概念ベースへの問合せ回数を減らして重心ベクトルを更新する。差分Δは、分かち書きした単語集合の差分であり、追加された単語の集合δ_+と削除された単語の集合δ_-とからなる。この場合、更新後の重心ベクトルG_q'は以下の式（２）によって、更新することができる。

【００８３】

【数２】

$$G_q' = \frac{m \times G_q + \sum_{v \in \delta_+ \cup \delta_-} v}{m + |\delta_+| - |\delta_-|} \quad \cdots\text{式（２）}$$

【００８４】

$|V_q|$が十分に大きければ（つまり、直前の検索文ｑが長ければ）、単語概念ベースの検索回数を大幅に削減することができ、検索速度が向上する。なお、上記式（２）において、V_q、δ_+、δ_-、はいずれも集合を示し、これらの絶対値記号は、当該集合に含

まれている要素の個数を示す。

【００８５】

上記各実施例は、コンピュータで使用可能なソフトウェアとして実施できる。プログラムは、ハードディスク、ＣＤ－ＲＯＭ、光記憶装置または磁気記憶装置等の任意のコンピュータ可読媒体に記憶できる。

【００８６】

実施例３によれば、直前の重心ベクトルと単語とを記憶し、直前の受信ベクトルと現在の受信ベクトルとの差分のみに応じて、次の重心ベクトルを更新するので、語の意味合いを考慮した逐次類似文書検索が高速で実行される。

【００８７】

上記実施例によれば、従来では不可能であった逐次類似文書検索を実現することができる。

【００８８】

実施例１では、文字単位ではなく、単語単位で検索を行うことによって、検索精度およびレスポンスタイムが低下せずに、データベースへの検索回数を下げて逐次類似文書検索を実現できる。また、実施例２では、概念ベース法を用いることによって、語の持つ豊かな意味合いを考慮した逐次類似文書検索を実現できる。さらに、実施例３では、索引語境界を検索する直前の重心ベクトルと単語集合を記憶して次の検索文ｑへの差分についてのみデータベースへ検索することによって、より高速に逐次類似文書検索を実現できる。

【００８９】

提案技術の特徴として、長い自然文を検索文ｑとする検索が高速に実現可能である。これによって、フリーフォームによる文書編集環境での逐次類似文書検索が可能になる。たとえば、「教えてｇｏｏ」に代表されるＱ＆Ａサイトにおいて、質問者が新しい質問文を作っている最中に、質問文を追記するにつれ、逐次的な検索によって似た質問文を即座に提示することができる。このように、似た質問を直ちに検索するので、その似た質問を行なうことを控えるであろうから、似た質問が繰り返されることを回避し、質問者・サイト運営者の双方にとってメリットが得られる。

【００９０】

また、上記各実施例における手段を工程に変更すれば、上記実施例を方法の発明として把握することができる。つまり、上記実施例は、逐次的類似文書検索手段が、類似文書を逐次的に検索し、記憶手段に記憶する逐次的類似文書検索段階と、上記逐次的類似文書検索段階で検索された検索結果を更新する更新段階とを有することを特徴とする逐次類似文書検索方法の例である。

【００９１】

この場合、検索語境界検索手段が、検索文の索引語境界を検出し、記憶手段に記憶する検索語境界検索段階を有し、上記逐次的類似文書検索段階は、上記索引語境界検出段階で上記検索文の索引語境界が検出される度に、索引語単位で逐次的に類似文書を検索する段階である。また、上記索引語境界検出段階は、局所性検知可能ハッシングを利用して、検索文の索引語境界を検出する段階である。さらに、上記索引語境界検出段階は、概念ベース法による類似文書検索を実現する段階である。

【００９２】

また、上記実施例をプログラムの発明として把握することができる。つまり、上記逐次類似文書検索方法をコンピュータに実行させるプログラムを想定することができる。そして、このプログラムを、半導体メモリ、ＣＤ、ＤＶＤ、磁気ディスク、光磁気ディスク、ＨＤ等、コンピュータ読み取り可能な記録媒体に記録するようにしてもよい。

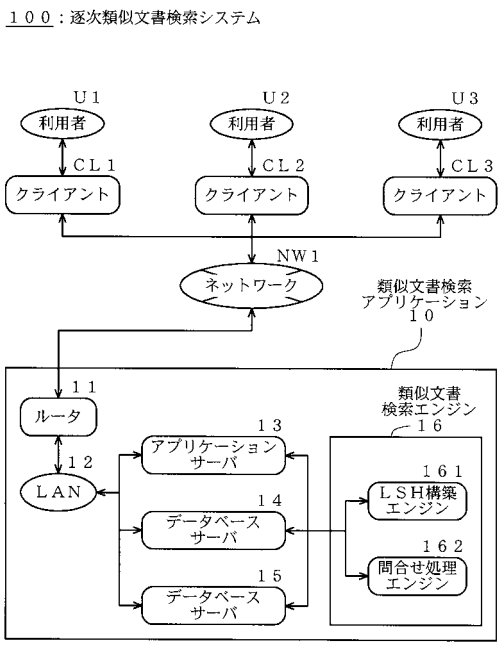
【符号の説明】

【００９３】

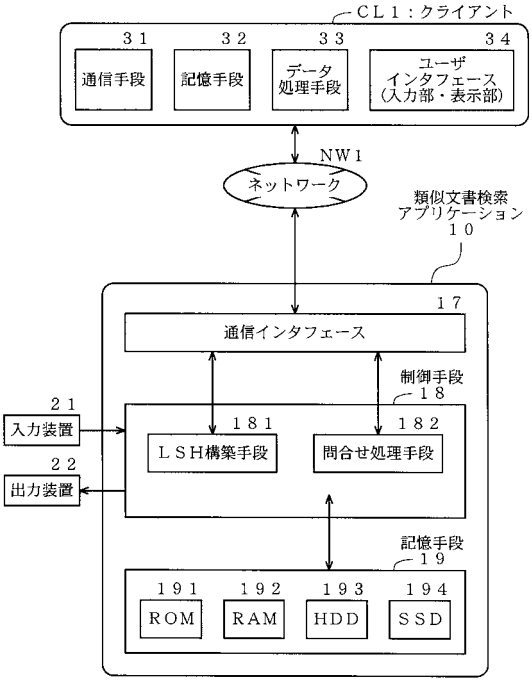
１００、２００、３００…逐次類似文書検索システム、
１０…類似文書検索アプリケーション、
１６…類似文書検索エンジン、

１８…制御手段、
１８１…ＬＳＨ構築手段、
１８２…問合せ処理手段、
ＤＢ１ああ文書データベース、
ＤＢ２…索引語データベース、
Ｄ１…元の文書集合、
Ｄ２…類似文書集合。

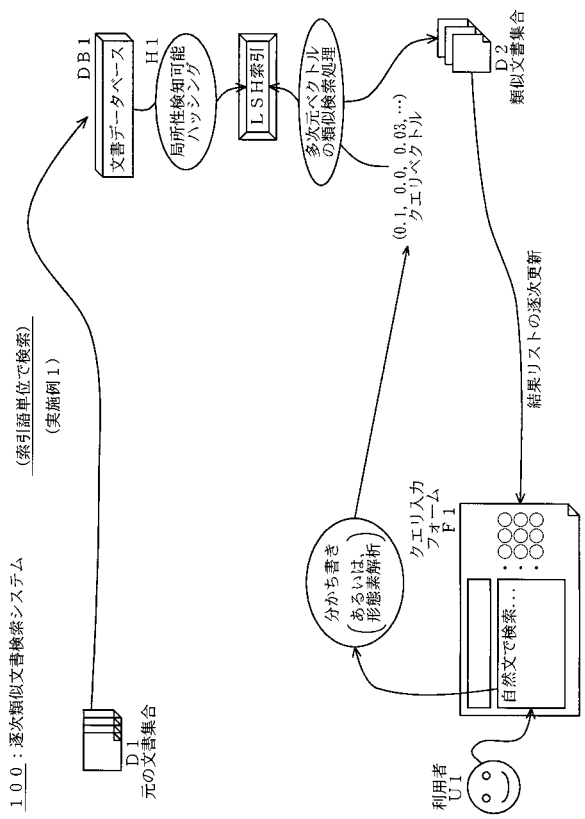
【図１】



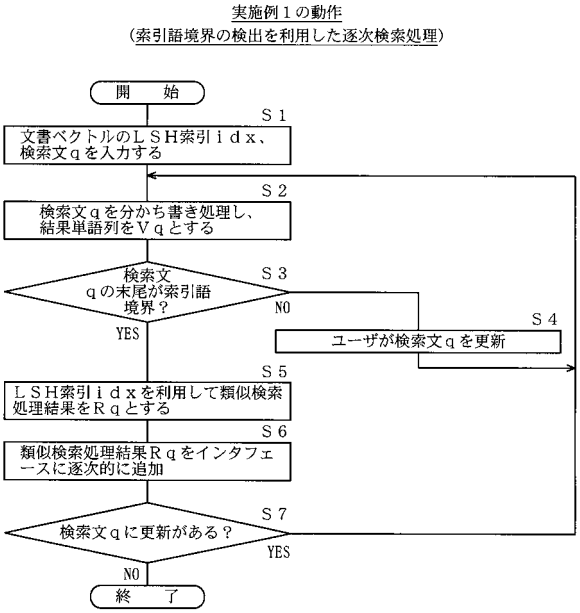
【図２】



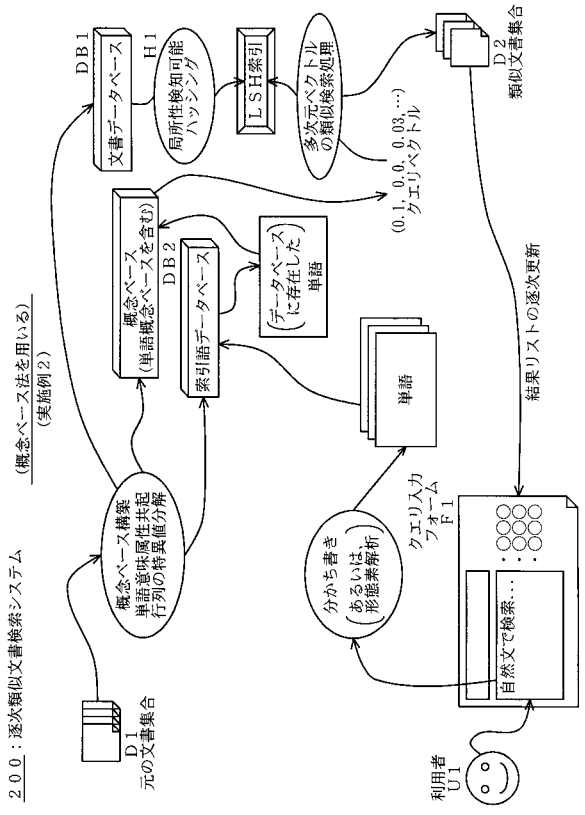
【図３】


１００：逐次類似文書検索システム
（索引語単位で検索）
（実施例１）
DB1
文書データベース
H1
局所性検知可能
ハッシング
LSH索引
多次元ベクトル
の類似検索処理
D2
類似文書集合
(0.1, 0.0, 0.03,...)
クエリベクトル
結果リストの逐次更新
クエリ入力
フォーム
F1
分かち書き
（あるいは
形態素解析）
自然文で検索…
D1
元の文書集合
利用者
U1

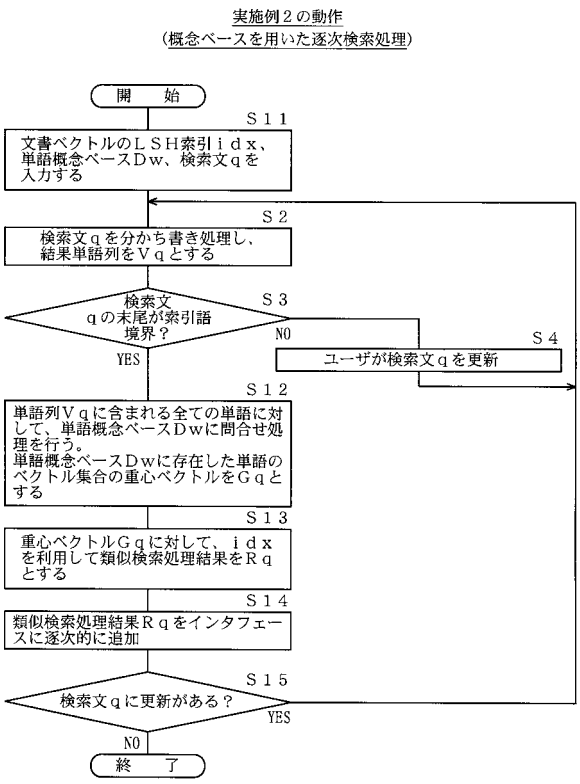
【図４】


実施例１の動作
（索引語境界の検出を利用した逐次検索処理）
開　始
S１
文書ベクトルのＬＳＨ索引ｉｄｘ、
検索文ｑを入力する
S２
検索文ｑを分かち書き処理し、
結果単語列をＶｑとする
S３
検索文
ｑの末尾が索引語
境界？
NO
YES
S４
ユーザが検索文ｑを更新
S５
ＬＳＨ索引ｉｄｘを利用して類似検索
処理結果をＲｑとする
S６
類似検索処理結果Ｒｑをインタフェ
ースに逐次的に追加
S７
検索文ｑに更新がある？
YES
NO
終　了

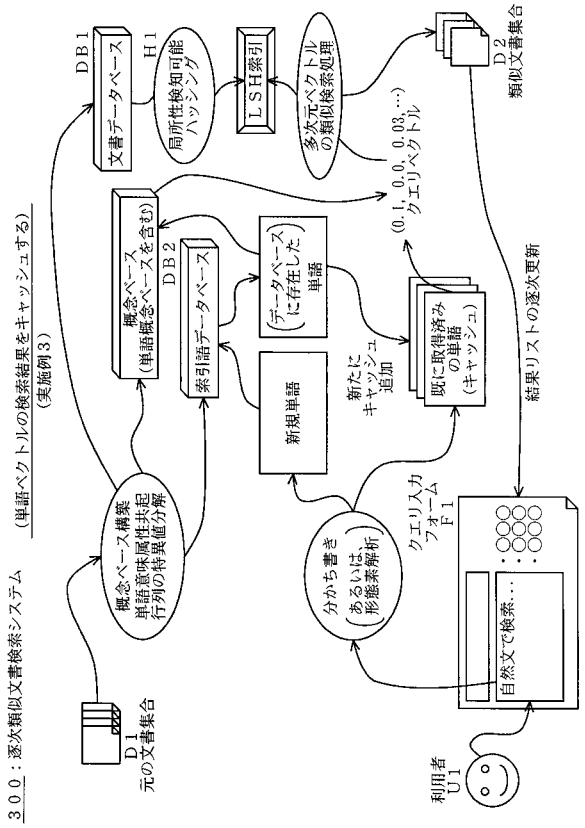
【図５】


２００：逐次類似文書検索システム
（概念ベース法を用いる）
（実施例２）
DB1
文書データベース
H1
局所性検知可能
ハッシング
LSH索引
多次元ベクトル
の類似検索処理
D2
類似文書集合
(0.1, 0.0, 0.03,...)
クエリベクトル
概念ベース
（単語概念ベースを含む）
DB2
索引語データベース
（データベース
に存在した
単語
単語
結果リストの逐次更新
概念ベース構築
単語意味展開性共起
行列の特異値分解
クエリ入力
フォーム
F1
分かち書き
（あるいは
形態素解析）
自然文で検索…
D1
元の文書集合
利用者
U1

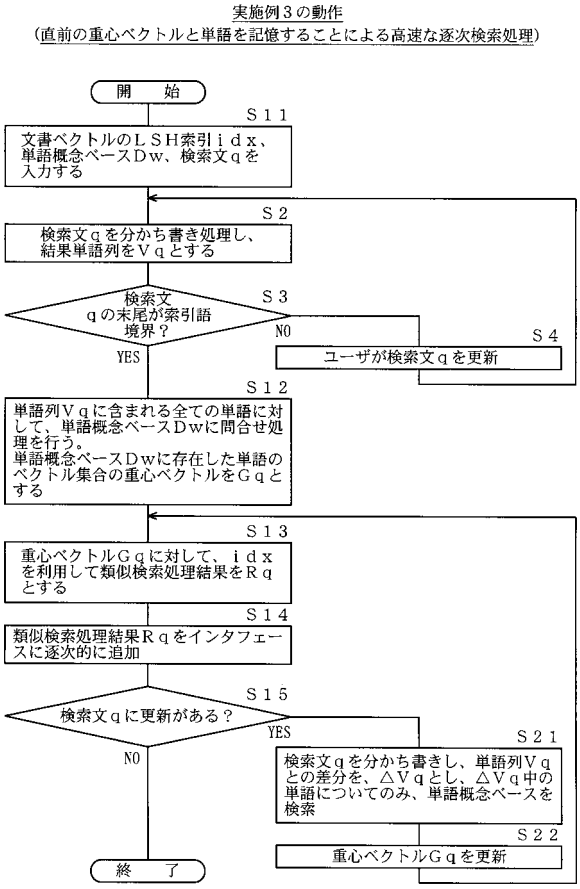
【図６】


実施例２の動作
（概念ベースを用いた逐次検索処理）
開　始
S１１
文書ベクトルのＬＳＨ索引ｉｄｘ、
単語概念ベースＤｗ、検索文ｑを
入力する
S２
検索文ｑを分かち書き処理し、
結果単語列をＶｑとする
S３
検索文
ｑの末尾が索引語
境界？
NO
YES
S４
ユーザが検索文ｑを更新
S１２
単語列Ｖｑに含まれる全ての単語に対
して、単語概念ベースＤｗに問合せ処
理を行う。
単語概念ベースＤｗに存在した単語の
ベクトル集合の重心ベクトルをＧｑと
する
S１３
重心ベクトルＧｑに対して、ｉｄｘ
を利用して類似検索処理結果をＲｑ
とする
S１４
類似検索処理結果Ｒｑをインタフェー
スに逐次的に追加
S１５
検索文ｑに更新がある？
YES
NO
終　了

【図７】


３００：逐次類似文書検索システム　（単語ベクトルの検索結果をキャッシュする）
（実施例３）
ＤＢ１
文書データベース
Ｈ１
局所性検知可能ハッシング
ＬＳＨ索引
多次元ベクトルの類似検索処理
Ｄ２
類似文書集合
(0.1, 0.0, 0.03,…)
クエリベクトル
概念ベース
（単語概念ベースを含む）
ＤＢ２
索引語データベース
（データベースに存在した）
単語
新規単語
新たにキャッシュ追加
既に取得済みの単語
（キャッシュ）
結果リストの逐次更新
概念ベース構築
単語意味属性共起
行列の特異値分解
分かち書き
（あるいは、
形態素解析）
クエリ入力
フォーム
Ｆ１
自然文で検索…
Ｄ１
元の文書集合
利用者
Ｕ１

【図８】


実施例３の動作
（直前の重心ベクトルと単語を記憶することによる高速な逐次検索処理）
開　始
Ｓ１１
文書ベクトルのＬＳＨ索引ｉｄｘ、単語概念ベースＤｗ、検索文ｑを入力する
Ｓ２
検索文ｑを分かち書き処理し、結果単語列をＶｑとする
Ｓ３
検索文ｑの末尾が索引語境界？
NO
Ｓ４
ユーザが検索文ｑを更新
YES
Ｓ１２
単語列Ｖｑに含まれる全ての単語に対して、単語概念ベースＤｗに問合せ処理を行う。
単語概念ベースＤｗに存在した単語のベクトル集合の重心ベクトルをＧｑとする
Ｓ１３
重心ベクトルＧｑに対して、ｉｄｘを利用して類似検索処理結果をＲｑとする
Ｓ１４
類似検索処理結果Ｒｑをインタフェースに逐次的に追加
Ｓ１５
検索文ｑに更新がある？
YES
Ｓ２１
検索文ｑを分かち書きし、単語列Ｖｑとの差分を、△Ｖｑとし、△Ｖｑ中の単語についてのみ、単語概念ベースを検索
NO
Ｓ２２
重心ベクトルＧｑを更新
終　了

フロントページの続き

(72)発明者　内山　匡
東京都千代田区大手町二丁目３番１号　日本電信電話株式会社内

Ｆターム(参考) 5B075 ND03　NK02　NK49　PR06　QM08