(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3552318号
(P3552318)
(45)発行日　平成16年8月11日(2004.8.11)　(24)登録日　平成16年5月14日(2004.5.14)

(51)Int.Cl.[7]　F I
G06F 17/30　G06F 17/30　414B
G06F 17/30　170A

請求項の数 6　(全 12 頁)

(21)出願番号　特願平7-2405
(22)出願日　平成7年1月11日(1995.1.11)
(65)公開番号　特開平8-190571
(43)公開日　平成8年7月23日(1996.7.23)
審査請求日　平成13年3月15日(2001.3.15)

(73)特許権者　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(74)代理人　100075096
弁理士　作田　康夫
(72)発明者　畠山　敦
神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内
(72)発明者　多田　勝己
神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内

最終頁に続く

(54)【発明の名称】文書検索方法およびシステム

(57)【特許請求の範囲】
【請求項１】
文書検索方法において、
記録装置に格納された文書ファイルから検索対象となる連接文字を含む文書ファイルの数を前記連接文字ごとに算出し、
算出した前記連接文字を含む文書ファイルの数が、しきい値より大きい場合には、先頭のビットから文書ファイルの番号に対応したビット列を用いて文書識別子情報を作成し、
算出した前記連接文字を含む文書ファイルの数が、しきい値より小さい場合には、前記連接文字が含まれる文書ファイルの番号のリストの形式で文書識別子情報を作成し、
作成した前記文書識別子情報を記録装置に格納し、
入力された検索語から連接文字を切り出し、
切り出した前記連接文字それぞれについて前記文書識別子情報を読み出し、
読み出した前記文書識別子情報の積集合を求めることにより、前記入力された検索語に含まれる連接文字を含む文書を抽出することを特徴とする文書検索方法。
【請求項２】
文書検索方法において、
記録装置に格納された文書ファイルから検索対象となる連接文字を含む文書ファイルの数を前記連接文字の種類ごとに算出し、
算出した前記連接文字を含む文書ファイルの数が、しきい値より大きい場合には、先頭のビットから文書ファイルの番号に対応したビット列を用いて文書識別子情報を作成し、

算出した前記連接文字を含む文書ファイルの数が、しきい値より小さい場合には、前記連接文字が含まれる文書ファイルの番号のリストの形式で文書識別子情報を作成し、
作成した前記文書識別子情報を記録装置に格納し、
前記文書識別子情報に基づいて文書検索を行うことを特徴とする文書検索方法。
【請求項３】
文書検索方法において、
記録装置に格納された文書ファイルから検索対象となる連接文字の出現頻度を前記連接文字の種類ごとに算出し、
算出した前記連接文字を含む文書ファイルの出現頻度が、しきい値より大きい場合には、先頭のビットから文書ファイルの番号に対応したビット列を用いて文書識別子情報を作成し、
算出した前記連接文字を含む文書ファイルの出現頻度が、しきい値より小さい場合には、前記連接文字が含まれる文書ファイルの番号のリストの形式で文書識別子情報を作成し、
作成した前記文書識別子情報を記録装置に格納し、
前記文書識別子情報に基づいて文書検索を行うことを特徴とする文書検索方法。
【請求項４】
文書検索方法において、
前記コンピュータは、連接文字の第一文字目を配列の要素に含む文字テーブルを記録装置へ格納し、
前記連接文字の第二文字目と前記連接文字を含む文書ファイルに関する情報である文書識別子情報を格納したファイルへのポインタ情報とを含むファイルポインタテーブルを記録装置へ格納し、
前記文字テーブルに含まれる連接文字の第一文字目と当該連接文字の第二文字目とを対応づけるように前記文字テーブルから前記ファイルポインタテーブルへのポインタ情報を前記文字テーブルに格納し、
指定された検索語に含まれる連接文字について、前記検索語に含まれる連接文字の第一文字目に対応づけられた前記文字テーブルのポインタ情報を参照し、
参照した前記ポインタ情報に基づいて前記検索語に含まれる連接文字の第二文字目を格納するファイルポインタテーブルを参照して前記検索語に含まれる連接文字を含む文書識別子情報を読み出すことを特徴とする文書検索方法。
【請求項５】
文書検索方法において、
記録装置に格納された文書ファイルを分割し、
前記分割された文書ファイルごとに検索対象となる連接文字の文書識別子情報を作成し、
前記分割された文書ファイルごとに作成した文書識別子情報をマージして記録装置に格納し、
格納した前記文書識別子情報に基づいて検索処理を行うことを特徴とする文書検索方法。
【請求項６】
文書検索システムにおいて、
前記文書検索システムは、記録装置に格納された文書ファイルから検索対象となる連接文字を含む文書ファイルの数を前記連接文字の種類ごとに算出する手段と、
算出した前記連接文字を含む文書ファイルの数が、しきい値より大きい場合には、先頭のビットから文書ファイルの番号に対応したビット列を用いて文書識別子情報を作成する手段と、
算出した前記連接文字を含む文書ファイルの数が、しきい値より小さい場合には、前記連接文字が含まれる文書ファイルの番号をリストの形式で文書識別子情報を作成する手段と、
作成した前記文書識別子情報を記録装置に格納する手段と、
前記文書識別子情報に基づいて文書検索を行う手段とを含むことを特徴とする文書検索システム。

【発明の詳細な説明】
【０００１】
【産業上の利用分野】
本発明は、文書データベースを、所定の文字列すなわち検索語を指定して文書の全文を対象として検索することにより、所望の文書を検索する文書検索方法に係わるものである。特に大量な文書を高速な検索を行う場合に好適な情報検索方法に関し、大規模文書データベースに適用されるものである。
【０００２】
【従来の技術】
先に、文書の登録の際にキーワード付けを行う必要のないフルテキストサーチ方式を特願平２－１９３０１５号（特開平３－１７４６５２号公報参照）で提案した。この方式は、文書を単語単位に圧縮した凝縮本文と、文書中の使用文字を一文字単位で登録した文字成分表を用いて、検索語に関連しない文書をふるい落とすことによってサーチ速度を等価的に高め、フルテキストサーチを実用レベルで高速に行うことを目的としたものである。また、この文字成分表を改良し更に高速なフルテキストサーチを実現する連接文字成分表方式を特願平３－３４２６９５号（特開平５－１７４０６４号公報参照）で提案した。この従来技術で用いる連接文字成分表は、テキストの中に含まれる所定の長さの連接する文字列を重複なく全て取り出し、これらを含む文書の識別子情報をビット列で記述するものである。しかし、全ての連接文字について識別子情報をビット列で記述すると、文字の組み合わせの個数分だけビット列が必要となり、連接文字成分表が膨大な容量になる。そこで、この従来技術では、ハッシュ関数を用いて１個のビット列に複数個の連接文字を割り当てるようにして、容量を抑える工夫をしている。
【０００３】
【発明が解決しようとする課題】
しかしながら、従来のハッシュ関数を用いて１個のビット列に複数個の連接文字を割り当てた場合には、同じビット列にまったく別の連接文字の文書識別子情報も重畳されることになる。従って、ある連接文字を指定して該当するビット列から文書識別子情報を取り出した場合、その情報からはまったく別の連接文字を含む文書が得られる可能性がある。つまり、ハッシュ関数を用いた連接文字成分表による検索結果には検索ノイズが含まれることになる。このことは、大量の文書を登録する大規模な文書検索システムでは、検索語に関連しない不要な文書のふるい落とし、すなわち絞り込みが適切に行われない可能性があることを意味し、その場合には検索性能の低下につながる。
【０００４】
ハッシュ関数を用いずに、全ての連接文字についてそれぞれ１個のビット列を対応させることも考えられるが、その場合にはビット列のデータ量が膨大なものとなり、実用的ではない。具体的に説明すると、日本語で使用する文字コードは、現在約８，０００種類あるので、２文字の組み合わせとしての連接文字の種類は、８，０００×８，０００＝６，４００万種類となる。登録する文書数を１００万件とした場合、この６，４００万種類のそれぞれの連接文字に１００万ｂｉｔの文書識別子情報を対応させるので、６，４００万種類×１００万ｂｉｔ＝８ＴＢｙｔｅもの容量が必要になる。この文字成分表の大きさに対し、文書本体の大きさを２０ＫＢ／件としても、１００万件で、２０ＫＢ×１００万件＝２０ＧＢｙｔｅであり、圧倒的に文字成分表の容量のほうが大きくなってしまう。
【０００５】
すなわち、本発明の解決しようとする課題は、大規模な情報検索システムにおいても検索ノイズの少ない連接文字成分表を、実用的な容量で実現することにある。
【０００６】
【課題を解決するための手段】
本発明は、以下の構成を採ることにより上述の課題を解決する。
【０００７】
文書のテキストデータにおける複数の文字の共起関係を記述した連接文字を連接文字ファ

イルに重複なく格納する連接文字格納ステップと、前記連接文字ファイルに格納された連接文字を参照して、指定した条件式中の検索語に含まれる連接文字を含む文書を検索結果の候補とする文書検索方法において、連接文字格納ステップとして、テキストデータ中に現れる連接文字成分の種類および各連接文字成分の出現する文書数を算出し、算出された文書数が所定のしきい値より大きい場合は該当文書の文書番号に対応する位置を“１”とするビット列として登録し、しきい値より小さい場合には該当文書の文書番号をバイナリデータとして格納することを特徴とする。

【０００８】

より詳細に言うと以下の（１）～（６）の各ステップに分けることができる。

【０００９】

（１）テキストデータ分割ステップ

（２）文書識別子情報作成ステップ

（３）文書識別子情報マージステップ

（４）検索語分割ステップ

（５）文書識別子情報探索ステップ

（６）文書識別子情報ＡＮＤステップ

（１）から（３）は文字成分表の登録のための処理であり、（４）から（６）はこれを用いた検索のための処理である。これより、各ステップの処理内容を説明する。

【００１０】

（１）テキストデータ分割ステップ

文字成分表への登録の際、文字の組合せの個数および各組合せに対応する文書識別視の記憶容量を抑えるために一回に処理する文書数を適切な数に分割する。分割する文書数は、予め設定してもよいし、登録に使用する計算機のメモリ容量から算出してもよい。

【００１１】

（２）文書識別子情報作成ステップ

（１）で分割した文書群のそれぞれについて別個に文書識別子情報を作成していく。具体的には、文書中に実際に現われた文字の組合せとその文字の組合せが現われた文書識別子の情報を対にして格納する。

【００１２】

（３）文書識別子情報マージステップ

（２）で作成した文書識別子情報を（１）で分割した文書群の数分マージして、登録文書全体の文字成分表を作成する。

【００１３】

（４）検索語分割ステップ

与えられた検索語を登録時と同じ方法で文字の組合せに分割する。

【００１４】

（５）文書識別子情報探索ステップ

（４）で分割した文字のそれぞれについて、文書識別子情報を探索する。

【００１５】

（６）文書識別子情報ＡＮＤステップ

（５）で得られた文書識別子情報のそれぞれについて、ＡＮＤ処理を行うことにより、与えられた検索語の全ての連接文字を含む文書を文字成分表サーチ結果として出力する。

【００１６】

【作用】

以下、これらのステップからなる本発明の文書検索方法の原理を説明した上で、その作用を説明する。

【００１７】

まず、本発明で用いる文字成分表の構成について説明する。本発明では、連接文字に対応する文書識別子情報を管理するのに、文字テーブル、ファイルポインタテーブルを用いる。図２は文字テーブルおよびファイルポインタテーブルの概要を示す図である。

【００１８】
たとえば、“構成”という文字列を含む文書を検索する場合には、まず文字テーブルについて“構”の文字に対応するレコードを参照してファイルポインタテーブルへのポインタ情報５８０を得る。次に、ファイルポインタテーブルの先頭から５８０バイト目からの各レコードを参照して、第二文字目が“成”のレコードを探索する。ファイルポインタテーブルには、各連接文字の第一文字目ごとに、先頭に第二文字目が０のレコードを格納しておく。第二文字目が０のレコードには、第一文字目の一文字を含んでいる全ての文書の文書識別子情報へのポインタを格納しておく。すなわち、第二文字目が０のレコードは、第一文字だけからなる単一文字に対応する文書識別子情報をアクセスするためのファイル識別子（以後ファイルＩＤとも呼ぶ）とファイル内バイト位置（以後オフセットとも呼ぶ）を格納する。したがって、各連接文字ごとに第二文字目が０のレコードが必ず存在するため、例えば、“構成”の連接文字を探索する場合は、“構”に対応するファイルポインタテーブルの先頭から５８０バイト目のレコードから探索を開始し、再び第二文字目が０になるまで探索を続け、もし“成”の文字が見つからない場合は、該当する連接文字がないと判断できる。図２の例では、“成”のレコードが存在するため、ここからファイルＩＤが１、オフセットが１０３４という文書識別子情報へアクセスするための情報を得ることができる。

【００１９】
文書識別子情報は、図３のように複数のファイルに分割格納する。ファイルポインタテーブルのファイルＩＤ情報により、どのファイルに文書識別子情報が格納されているかを特定する。なおかつ特定のファイルＩＤは、文書識別子情報をビット列で持つとあらかじめ決めておく。図３の例では、ファイル１が文書識別子情報をビット列で持つファイルとしている。図２の例では、連接文字“構成”に関する文書識別子情報へのアクセス情報として、ファイルＩＤが１、オフセットが１，０３４が得られる。したがって、ファイル１内の１，０３４バイト目からのビット列“０１１１０１０１０１．．．．”が文書識別子情報として得られることになる。このビット列は、先頭ビットから文書番号に対応して、“１”が連接文字“構成”を含む文書を示すことになる。すなわち、この例では、“構成”を含む文書の文書番号は、１、２、３、５、７、９．．．．となる。図３の他のファイル（ファイル２及びファイル３）は文書識別子情報をＩＤリストの形式で格納したものである。各ＩＤリストの先頭は格納してある文書番号の個数を示している。例えば、連接文字“構造”の場合、図２の例では、ファイルＩＤが２、オフセットが３４０であるので、ファイル２の先頭から３４０バイト目を参照することによって、連接文字“構造”を含む文書数が５６個あり、文書番号が５６２、１０３８、．．．であることがわかる。

【００２０】
このように、ファイルポインタテーブルには、データベース中に存在する連接文字のみを登録するので、データベース中に存在しない文字の組み合わせは全て排除できるという利点がある。したがって、文字テーブルやファイルポインタテーブルで実現している連接文字の管理情報を格納するファイル量やメモリ量を大幅に削減することができる。また、文書識別子情報をビット列あるいはＩＤリストの形式で格納し、多くの文書を格納する場合はビット列で、少ない文書を格納する場合はＩＤリストの形式で管理することによりファイル容量を大幅に削減することができる。具体的に説明すると、ビットリストの形式で文書識別子情報を格納するには、常にデータベースに登録した全件分のビット数が必要になるが、ＩＤリストの形式で文書識別子情報を格納する場合には、文書識別子を表わすビット数×登録文書数ですむことになる。例えば、データベースの全登録件数が１００万件で、一個の文書識別子情報を表わすのに３２ビットを割り当てるとすると以下の格納領域が必要となる。連接文字“構造”を含む文書を１０件登録する場合に、ビット列ならば、１００万ｂｉｔ＝１２５ＫＢの格納領域が必要となるが、ＩＤリスト形式ならば、３２ｂｉｔ×１０件＝４０Ｂの格納領域ですむことになる。一方、例えば、連接文字“構成”を含む文書が１００万件中で９０万件ある場合には、ビット列ならば、１００万ｂｉｔ＝１２５ＫＢの格納領域にすむのに対し、ＩＤリスト形式の場合、３２ｂｉｔ×９０万件＝３．

６ＭＢの領域が必要となる。したがって、この１００万件を、文書識別子３２ビットで格納する場合には、１００万ｂｉｔ÷３２ｂｉｔ＝３１，２５０件を境として、これよりも登録件数が多い場合はビット列形式で、少ない場合はＩＤリスト形式で文書識別子情報を格納するのが、最も格納領域を有効に使用する方法である。

【００２１】

次に、このような文字成分表の登録の方法について、原理を説明する。文字テーブルとファイルポインタテーブルを用い、データベース中に用いられる連接文字のみを文字成分表に登録することにより、ファイル容量を実用容量に抑えることができることは既に説明した。

【００２２】

したがって、登録時に全ての連接文字成分について管理をしようとすると、メモリ容量が足りなくなり、文字成分表を作ることが不可能となる。磁気ディスクをワークにして情報を一旦退避する方法もあるが、アクセス速度が遅いので登録処理に極めて時間が掛かることになる。そこで、図４のように登録するテキストデータを分割して、分割したテキストデータ毎に文字成分表を作成し、最後にこれらをマージして全テキストデータの文字成分表を作成する。図４では、全部で２万４千件のテキストデータを８千件毎に分割して文字成分表を作成する例を示している。“構成”という連接文字について、最初の８千件のテキストデータでは、文書番号５０、１４５、２９０．．．．が文書識別子情報として蓄えられる。同様に、次の８千件、その次の８千件についても各分割したテキストデータ毎に文字成分表を作成する。最後に、それぞれで得られた文書識別子情報をマージして、本図の例では、“構成”の連接文字に対する文書識別子情報として、５０、１４５、２９０、８０９６、１２３６５、１７８５１、２２９８９．．．という情報を作成する。

【００２３】

検索の際には、入力された検索語を連接文字に分割し、それぞれの連接文字に対応する文書識別子情報を読み出してきて、それらの情報の積集合を取り、これを文字成分表の検索結果とする。すなわち、“建造物”という検索語については、“建造”と“造物”の２種類の連接文字について、それぞれ文字成分表の文書識別子情報を読み出してそれらの積を演算する。例えば、連接文字“建造”に対応する文書識別子情報が５６２、１０３８、２４５８．．．．で、連接文字“造物”に対応する文書識別子情報が２６１、５６２、２４５８．．．．の場合は、検索語“建造物”の文字成分表サーチ結果は文書番号で５６２、２４５８．．．．となる。

【００２４】

このように、各連接文字に対する文書識別子情報はノイズのない情報であるため、これらの文書識別子情報を論理式演算（ＡＮＤ）して得られる文字成分表サーチ結果も、従来のハッシングを行う文字成分表のサーチ結果に比べ、ハッシングに起因するノイズが除去されることになり、検索精度が大幅に向上できることになる。

【００２５】

【実施例】

以下、本発明の実施例について図を用いて詳細に説明する。

【００２６】

図１は、本実施例の構成を示す図である。本実施例は、登録検索用の端末１０１、１０２、．．．１１０、ネットワーク２００、文書サーバ１０００からなる。文書サーバ１０００には、ＬＡＮアダプタ１０１０、ＣＰＵ１０２０、ワークメモリ１０３０、文字テーブル１１００とファイルポインタテーブル１２００を格納するメモリ、テキストデータ分割プログラム１３１０、文書識別子情報作成プログラム１３２０、文書識別子情報マージプログラム１３３０、検索語分割プログラム１３４０、文書識別子情報探索プログラム１３５０、文書識別子情報ＡＮＤプログラム１３６０を格納するメモリ、文字成分表を分割して格納するファイル１４０１、１４０２、．．．、テキストデータ１４１０からなる。

【００２７】

データの登録時には、テキストデータ分割プログラム１３１０で登録する文書データを一

定の件数に分割し、分割したそれぞれのテキストデータについて文書識別子情報作成プログラム１３２０で文書識別子情報を作成して、最後に分割して作成したそれぞれの文書識別子情報を文書識別子情報マージプログラム１３３０でマージして文字テーブル１１００、ファイルポインタテーブル１２００、文字成分表１４０１、１４０２、１４０３を作成する。

【００２８】

また、データの検索時には、各端末から与えられた検索語を検索語分割プログラム１３４０によって文書識別子情報を作成したときと同じアルゴリズムで連接文字に分割し、それぞれの連接文字について文書識別子情報探索プログラム１３５０で該当する文書識別子情報を文字成分表１４０１、１４０２、１４０３から取り出す。そして、検索語を構成する全ての連接文字に対応する文書識別子情報を文書識別子情報ＡＮＤプログラム１３６０によってＡＮＤすることで検索語を含む文書を文字成分表のサーチ結果とする。

【００２９】

まず、データの登録処理に従い、文字成分表の作成手順を説明し、次に検索処理に従って文字成分表による候補文書の抽出過程を説明する。作用の項でも説明したように、大量の文書について文字成分表を一度に登録するには、大量のメモリを使用しなければならないので、本実施例では８，０００件ごとに小さな文字成分表を作成し、最後に一つの文字成分表に統合する処理を行う。図５に、この文書識別子情報作成処理の手順を示す。まず、８，０００件のそれぞれの文書について（５０１０）連接文字の抽出（５０２０）を行い、切り出した連接文字についてその出現頻度情報を計数（５０３０）する。そして、算出した出現頻度にしたがって文書識別子情報を格納するメモリエリアをワークメモリ上に確保し、それぞれの連接文字の出現頻度情報が所定のしきい値より大きい場合にはビット列で各連接文字が出現する文書番号を文書識別子情報として登録（５０４０）していく。８，０００件の全ての文書について文書識別子情報を登録し終わったらファイルに文字テーブル、ファイルポインタテーブル、文書識別子情報を格納（５０５０）しメモリ領域を解放する。８，０００件単位にこのように小さな分割文字成分表を作成し、最後に各分割文字成分表をマージ（５０６０）してデータベース全体の文字成分表を作成する。

【００３０】

この分割文字成分表のマージ処理（５０６０）は、図６に示すとおり各分割文字成分表の文字テーブルとファイルポインタテーブルを参照し、それぞれの連接文字に対応する文書識別子情報を統合する形で進めていく。図６は二個の分割文字成分表を一個の文字成分表に統合する例を示している。具体的な処理の手順を図７に示す。まず、それぞれの分割文字成分表の文字テーブルを参照（７０１０）し、統合した文字テーブルを作成（７０３０）する。この時文字テーブルの各レコードについて（７０２０）、どちらか一方にしか登録されていないレコードについては、登録されている側に記録されているファイルポインタテーブルの各文字について（７０４０）内容を統合したファイルポインタテーブルに登録する（７０５０）とともに、ファイルポインタテーブルで管理されている文書識別子情報をマージ前の小さな文字成分表からマージ後の文字成分表へコピー（７０６０）していく。また、双方の文字テーブルに同じ文字が存在する場合には、記録されているファイルポインタテーブルの各文字について（７０７０）、ファイルポインタテーブルに記載された第二文字目を比較しながら統合したファイルポインタテーブルを作成（７０８０）していく。すなわち、ファイルポインタテーブルの第二文字目が一致しない場合には、該当する文書識別子情報をコピー（７０９０）し、一致する場合には双方の文書識別子情報をマージ（７１００）して格納する。

【００３１】

この文書識別子情報のマージ及びコピーの際には、マージ後の登録件数から所定の件数よりも多い場合にはビット列に、少ない場合にはＩＤリストの形式にして格納する。

【００３２】

以上のマージ処理アルゴリズムを図６を用いて具体的に説明する。“構”の文字は文字テーブル１および文字テーブル２のどちらにも存在する。したがって、“構”の文字に対応

するファイルポインタテーブル１の内容とファイルポインタテーブル２の内容を統合ファイルポインタテーブルに登録していく。ファイルポインタテーブルにおける該当レコードの先頭の第二文字目が０のレコードは、“構”の一文字を含む文書の識別子情報をアクセスするための情報を格納している。この第二文字目が０のレコードはファイルポインタテーブル１とファイルポインタテーブル２の両方に存在するので、双方のファイルＩＤとオフセットで与えられる文書識別子情報をマージして統合文字成分表に登録する。“構”に対応するファイルポインタテーブルの第２レコード“成”についても同様である。第３レコードについてはファイルポインタテーブル１が“造”であるのに対して、ファイルポインタテーブル２では“築”と異なっている。したがって、それぞれの文書識別子情報をマージ前の小さな文字成分表から統合文字成分表へコピーする。

【００３３】

検索処理は、図８に示す手順で行う。まず、検索語から連接文字を切り出す（８０１０）。次に、切り出したそれぞれの連接文字について（８０２０）、文字テーブルを探索する（８０３０）。そして、該当するファイルポインタテーブルの各レコードについて、第二文字目の探索を行い（８０４０）該当するファイルＩＤとオフセットを得る。こうして、得られた文書識別子情報を格納したファイルとそのオフセット値より、該当する連接文字に対応するＩＤリストまたはビット列を読み出し、ＩＤリストの場合にはこれをビット列に変換することにより文書識別子情報を取得する（８０５０）。この文書識別子情報の取得の過程で該当する連接文字が文字成分表に登録されていない場合（８０６０、８０７０）には、すなわち検索語を構成する連接文字のうちどれか一つでも文字成分表に登録されていなければ、検索語を含む該当文書がないということを意味することになるため検索結果として０件という結果を、文書識別子情報探索プログラム１３５０がＬＡＮアダプタ１０１０を介して検索端末に返す。

【００３４】

検索語を構成する全ての連接文字について該当する文書識別子情報が得られた場合は、得られたそれぞれの文書識別子情報の積集合をとることによって、指定された検索語中の全ての連接文字を含む文書のみを抽出することができる。

【００３５】

このようにして得られた文字成分表の検索結果は、検索ノイズが非常に少ないので、文字成分表のサーチ結果を表示しても十分実用できる。もちろん、文字成分表のサーチ結果をもとに、文書本文を検索し実際に検索語を含む文書のみに絞り込むかあるいは、複数の検索語間の位置的関係を満たす文書を探すことも可能である。また、文字成分表の検索結果を一度検索端末に表示し、ユーザの指定により本文の探索を行うかどうかを決定してもよい。

【００３６】

以上、本実施例によれば、データベース中に存在する連接文字のみを登録するので、データベース中に存在しない文字の組み合わせは全て排除できるという利点がある。また、文書識別子情報をビット列とＩＤリストの形式で格納し、多くの文書識別子情報を格納する場合はビット列で、少ない文書識別子情報を格納する場合はＩＤリストの形式で格納することでファイル容量を大幅に削減することができる。

【００３７】

さらに、各連接文字に対して文書識別子情報は必ずその連接文字を含むノイズのない情報であるから、これらの文書識別子情報をＡＮＤして得られる文字成分表サーチ結果も、検索精度を大幅に向上することができる。

【００３８】

また、本発明によれば２文字以上の連接文字についても登録することにより、さらに文字成分表サーチの検索ノイズを少なくすることも可能である。

【００３９】

【発明の効果】

本発明によれば、文書識別子情報をビット列とＩＤリストのどちらかの形式で選択的に格

納することにし、多くの文書識別子情報を格納する場合はビット列で、少ない文書識別子情報を格納する場合はＩＤリストの形式で格納することでファイル容量を大幅に削減することができる。

【００４０】

また、各連接文字に対して文書識別子情報は必ずその連接文字を含むノイズのない情報であり、これらの文書識別子情報を検索語の連接文字の個数分ＡＮＤするので、文字成分表サーチの検索精度を大幅に向上することができる。これにより、検索語間の位置的な条件などを検索する場合にも、より本文情報の検索範囲を狭めることができるという利点がある。

【００４１】

さらに、文字テーブル及びファイルポインタテーブルを用いることにより、データベース中に存在する連接文字のみを登録するので、データベース中に存在しない文字の組み合わせは全て排除できるので、連接文字を管理するために必要なメモリ量を少なくできるという利点がある。

【００４２】

さらにまた、文字成分表の登録の際に、登録文書を分割して小さな分割文字成分表を作成し、後でこれらの分割文字成分表をマージして目的の文字成分表を作成することにより、少ないメモリ容量でも効率的に大きなデータベースの文字成分表を作成することができる。

【図面の簡単な説明】

【図１】本発明の第一の実施例の構成図である。

【図２】文字成分表のテーブル構成図である。

【図３】文書識別子情報格納ファイルの概要を示す図である。

【図４】文字成分表登録処理の概要を示す図である。

【図５】登録処理の流れを示すＰＡＤ図である。

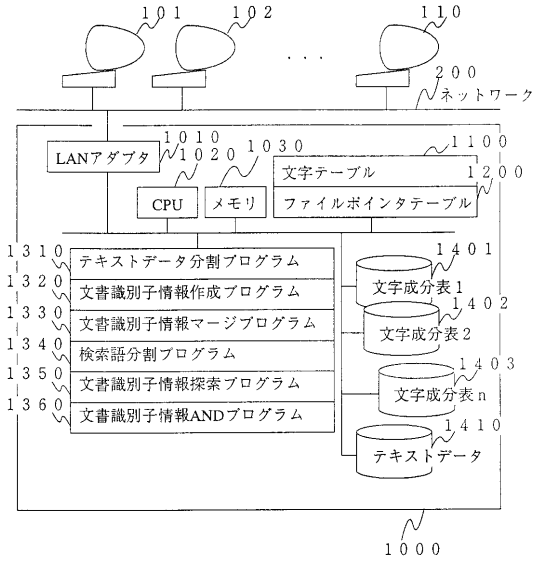
【図６】分割文字成分表の統合処理を示す概念図である。

【図７】統合処理の流れを示すＰＡＤ図である。

【図８】検索処理の流れを示すＰＡＤ図である。

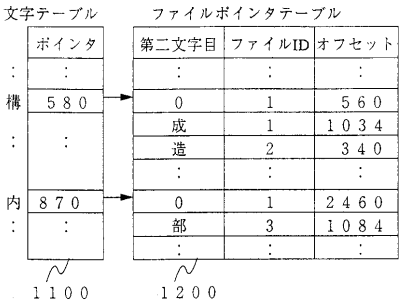
【図１】

図１



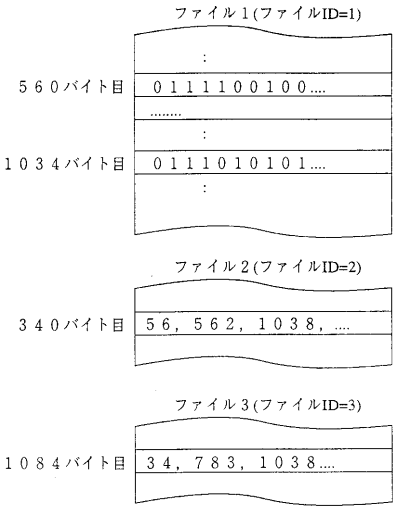
【図２】

図２



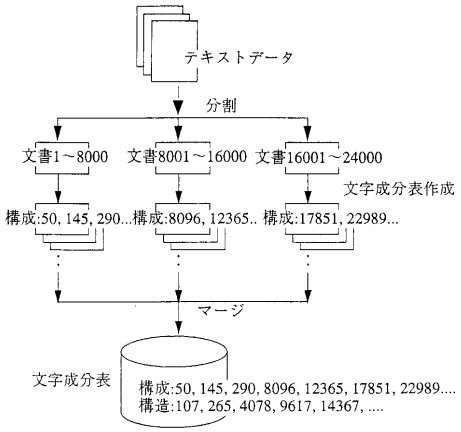
【図３】

図３



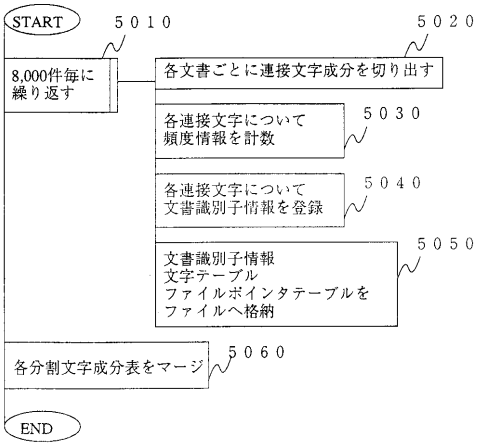
【図４】

図４



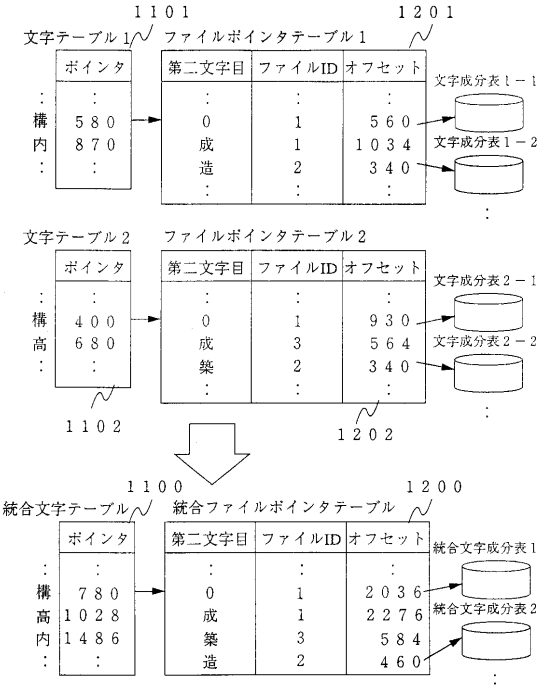
【図５】

図５



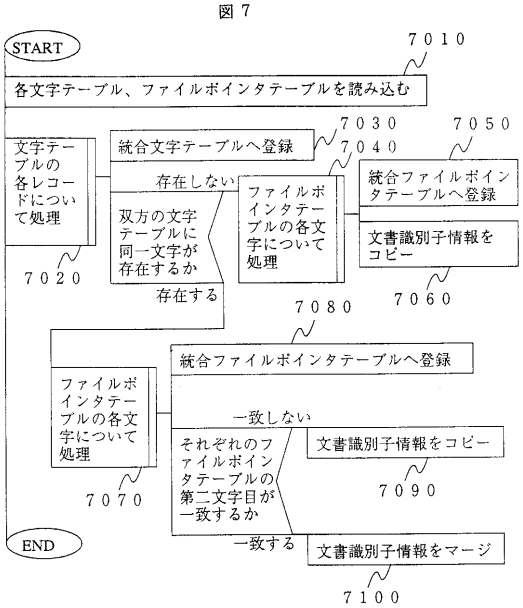
【図６】

図６



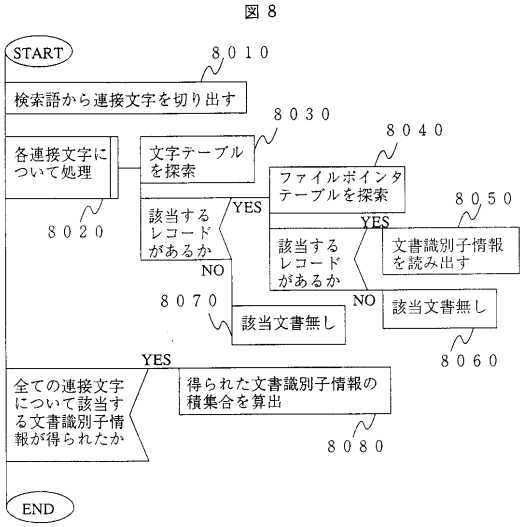
【図７】

図７



【図８】

図８



フロントページの続き

(72)発明者 加藤 寛次
神奈川県川崎市麻生区王禅寺１０９９番地 株式会社日立製作所 システム開発研究所内
(72)発明者 浅川 悟志
神奈川県横浜市戸塚区戸塚町５０３０番地 株式会社日立製作所 ソフトウエア開発本部内

審査官 田川 泰宏

(56)参考文献 特開平０４－２７４５５７（ＪＰ，Ａ）
特開平０５－１７４０６４（ＪＰ，Ａ）
特開平０７－３１９９２０（ＪＰ，Ａ）
特開平０８－０３０６３３（ＪＰ，Ａ）
岩崎雅二郎,小川泰嗣，文字成分表による文字列検索の実現と評価，情報処理学会研究報告(93-DBS-92)，１９９３年 3月２２日，Vol.93,No.29，p.1-10
小川泰嗣,岩崎雅二郎,林大川，全文検索のための文字成分表方式の改良，情報処理学会研究報告(94-DBS-99)，１９９４年 7月２２日，Vol.94,No.62，p.261-264
畠山敦，ソフトウェアによるテキストサーチマシンの実現，情報処理学会研究報告(92-FI-25)，１９９２年 5月１２日，Vol.92,No.32，p.19-25

(58)調査した分野(Int.Cl.^{7}，ＤＢ名)
G06F 17/30
JICSTファイル(JOIS)