5D-7

SubstringArray を用いた WWW 大規模全文検索
Scalable Full Text Search Using SubstringArray

内藤 一兵衛 † **三上 啓太** † **上田 和紀** ‡
Ichibe NAITO Keita MIKAMI Kazunori UEDA

1. はじめに

WWW は膨大に膨れ上がり, 日々新しい知識が作り出されている．この知識を有効に使うため，全文検索システムが必要とされている．また，200GB を超えるディスクが普及した現在，一台のマシンで処理するデータ量も増え索引を見直す必要がある．

日本語など分かち書きされていない言語の検索には N-Gram や形態素解析で単語に分割することが多い．これらの方法は結果を返すために単語の検索結果同士でインターセクションを取る必要がある．インターセクションのコストは $O(N)$ だが，大規模な文書を対象とした場合には向かない．ページに対するスコア順に上位数件のページのみ検索することもできるが，検索結果の統計的な処理も行いたいという要求もある．

これに対し，文字列を木構造で管理する索引はインターセクションを必要としないことから大規模な文書群に対して有効だと考えられる．

そこで，本研究では大規模な WWW 全文検索システムでの使用を目的として, ディスクベースの SuffixArray を改良した索引を提案する．また，実験を通して 1 台のマシンで全文検索が行える規模について考察する．

2. SuffixArray

SuffixArray とは全文検索の索引の一つで，対象とする文章の全ての文字を接尾語 (suffix) の始点として, それら接尾語を辞書順にソートする．接尾語の始点を配列として持ったのが SuffixArray である．これを二分探索することにより検索文字列を探すのがこの索引の仕組みである．元文書がメモリにのる範囲では高速に検索が出来るが，ディスクベースの研究は多くない．

3. SubstringArray

SubstringArray とは大規模な WWW 全文検索において検索結果全件を高速に取得することを目的として SuffixArray を大規模に対応させたものである．検索キーワードの多くは 10 文字程度の文字列で文字列長が長くなるほど検索ヒット件数は少ない傾向にある．このことから，完全な SuffixArray を構築する必要はない．SuffixArray は文章の全てのポイントから文末までを検索対象とするが，検索キーワードが短いことを考慮すると，Suffix ではなく部分文字列 (substring) でも良いと考えられる．部分文字列の長さ L は構築前に設定する．

†早稲田大学大学院理工学研究科 情報・ネットワーク専攻
‡早稲田大学理工学部 コンピュータ・ネットワーク工学科

SubstringArray は図 1 に示すように位置情報列と Substring 列から構成される．位置情報とは元文書へのポインタである．

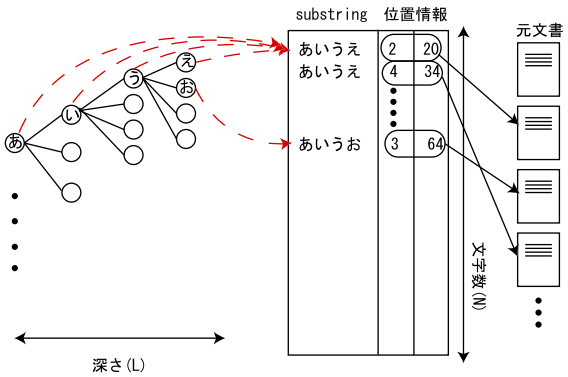

図 1: SubstringArray の構成

3.1 構築

構築は以下の手順で行う．

1. インデックスポイントの切り出し
   インデックスポイントとは「テキスト中の検索可能な位置」のことで, 文書中に現れたタグ以外の部分全てに割り当てる．

   SuffixArray の構築においては元文書とポインタを別々に管理することが多い．しかし, 今回は大規模かつ複数の文書を対象とするため, 元文書, ポインタ両方をメモリ上に載せることはできず, 元文書にアクセスすることはランダムアクセスの増大につながってしまう．

   そこで, 割り当てたポイントを位置情報ファイルに書き込む際に，同時にその場所から L 文字を Substring 列として書き出す．これによりメモリに乗らない範囲での Substring のソートが可能となる．

2. インデックスポイントのソート
   切り出したインデックスポイントをポインタ先の文字列をキーに辞書順にソートする．メモリに載る範囲ごとに Multi-key Quicksort[1] を行い, それらをマージソートしていく．

検索キーワードの長さが L を超えない限り, インターセクションの必要はない．作成途中の索引サイズは位置情報を 8 バイトで元文書 N に対して $(2L+8)N$ バイトとなる．

表 1: SubstringArray(large) の解析

階層	1	2	3	4	5	6	7	8	9
平均子ノード数	598.22	15.36	4.44	2.34	1.65	1.34	1.13	1.11	1.08
平均データ数	1438.57	86.97	18.23	7.25	4.09	2.82	2.19	1.85	1.64

表 2: SubstringArray の構築

データ (ページ数)	data size (GB)	日本語文字数	切り出し時間 (h:m:s)	索引サイズ) (GB)	ソート時間 (L=10) (h:m:s)
mid(2,000,000)	20	1,338,344,913	1:29:34	43	5:26:43
large(10,000,000)	100	6,721,495,409	8:02:39	215	33:10:22

3.2 追加, 更新, 削除

SuffixArray の場合は文書の一部を変更するとその後の suffix の位置情報が全てずれてしまい，これらの補正には多大のコストがかかることが問題であった．

SubstringArray では，追加，更新，削除は全て元文書単位でおこなう事で位置情報の補正を回避する．対象文書は十分に小さいため，更新のコストも大きくない．

追加の場合, 追加文書のみで部分文字列の切り出し, ソートを行い生成した索引と主索引とをマージすることで, 文書の追加を行うことができる．削除の場合は対象文書 id を位置情報ファイルをスキャンして該当 id を含むデータを消せばいい．更新は削除と追加を同時に行えば１回のスキャンで行える．

3.3 検索

SubstringArray はメモリに載らないサイズで作成されているため，SuffiArray 同様ディスク上の配列を二分探索するのが簡単な方法である．しかし, この方法ではメモリを有効に使えていない．

構築実験で作った large の SubstringArray を解析した結果を表 1 に示す．最初の数文字で子ノードの平均が減っている．それまでの探索を工夫すれば探索時間が効率化できることがわかる．そこで，先頭の N バイトを配列でメモリ上に保持しておくことで高速に探索範囲を狭めることができる．

本研究では先頭 2 文字目までをメモリ上に保持し，そこからディスクの二分探索を行う．

4. 実装・評価

4.1 構築実験

実験は WEB クローラーで集めた HTML 文書を対象に行った．対象とするデータ規模は JP ドメインページ 200 万，1000 万ページである．実験マシンは Opteron1.4GHz デュアルマシンで，メモリは 4GB である．HDD は 7200rpmSATA キャッシュ8MB の 250GB，4 台をソフトウェア RAID0 でつないでいる．Substring の長さ L は 10 文字である．

実験結果は表 2 に示す．現実的な時間で構築できている．

4.2 検索実験

上記で作成したデータに対し検索を行った．200 万ページが mid，1000 万ページが large である．クエリーセットは 600 単語，2 バイト文字のみで構成され, 文字列長は 1〜10 文字まで 60 単語ずつである．SubstringArray においてディスクキャッシュの影響を防ぐため，先頭 2 文字が同じものは連続しない．実験では，目的のキーワードを見つけるまでの平均 disk read 回数とディスク上のヒット位置を調べるまでにかかる時間を計った．

実験結果を表 3 に示す.

表 3: 検索時間

対象	先頭配列 (文字数)	平均 disk read 回数	平均検索時間 (msec)
mid	0	18.67	89.8
mid	1	8.91	84.7
mid	2	6.74	68.7
large	0	18.71	125.6
large	1	9.04	102.5
large	2	7.01	100.8

実験結果から，20GB1 から 100GB になっても検索時間は増加は少ない．検索結果をディスクからメモリに移すコストは線形時間であるため, 大量の結果がある場合は 1 秒を超えてしまうが，結果が 10 万件以下であれば検索時間は 0.3 秒程度で収まっている．

5. おわりに

大規模なデータを対象としたときに，一度に全結果を取得する目的で SubstringArray を提案した．また，この索引の利点としてキャッシュに SubstringArray の始点, 終点さえ覚えておけば索引を引きなおす必要はない．

今後の課題としては, 検索時にメモリ上に保持する木構造の改良によりより多くの Substring を保持できるよう改善したい．

参考文献

[1] Jon Bentley and Robert Sedgewick. Fast Algorithms for Sorting and Searching Strings. In Proc. 8th ACM-SIAM Symposium on Discrete Algorithm, pp-360-369, 1997

[2] Udi Manber, Gene Myers, Suffix Arrays: A New Method for On-line String Searches, In Proc. 1st ACM-SIAM Symposium on Discrete Algorithms, pp. 319-327, 1990.