5T－4

高速全文検索のための フレキシブル文字列インバージョン法(1) 方式概要

福島 俊一 赤峯 享
NEC 情報メディア研究所

1 はじめに

インターネットの爆発的な普及やイントラネット構築ブームを背景に、大規模で多様なテキスト情報の効率よい管理・検索に対するニーズは非常な高まりを示している。対象となるテキストは新しい話題・用語を次々に取り込んで増加し、幅広い層のユーザが多様な視点から検索することを考えると、テキストをあらかじめ少ないキーワード集合で抽象化しておくアプローチでは限界がある。テキスト全文をそのまま検索対象とし、大規模なテキスト集合に対する高速な全文検索を実現する技術が望まれている。

専用ハードウェアを用いようと大規模テキストの全文走査はもはや現実的でない。したがって、高速全文検索の実現にあたっては、インバーテッドファイルの設計が重要なポイントになる。インバーテッドファイルの形式は、(1)キーとしてテキストから単語を取り出すか(=1a)/文字列を取り出すか(=1b)、(2)キーに対応づける位置情報をテキストIDのみとするか(=2a)/テキストID＋オフセット(テキスト内位置)とするか(=2b)、の組み合わせにより4通りに大別できる。

キーとして単語を取り出すタイプ(1a×2a)(1a×2b)では、形態素解析誤りや未知語に起因するキーの登録洩れや誤登録が発生する(結果として検索洩れを生む)。また、キー文字列にテキストIDを対応づけるタイプ(1b×2a)は、検索条件語を複数の部分文字列に分解してインバーテッドファイルと照合するだけでは検索ノイズ(過剰ヒット)が避けられない[1]。後処理として全文走査を実行すれば検索ノイズは除去できるが、その場合、全文走査による検索レスポンスの低下と、オリジナル文書形式と別にプレインテキストも保存せねばならない運用上のオーバヘッドが発生する。

本論文で提案するフレキシブル文字列インバージョン法(以下ではFSI法と略す)は、(1b×2b)タイプに属する。FSI法では、検索レスポンスの高速化のために位置情報データの読み出し量を削減する方針をとり、文字列統計に基づくキー文字列長の設定と縮退文脈の付与、位置情報データの圧縮手法などを導入した点が特長である。

2 検索高速化のための着眼点

キー文字列にテキストID＋オフセットを対応付けるタイプ(1b×2b)のナイーブな実装例を図1に示す。この例では、文字列長1の各キーに対して、その文字の出現位置をインバーテッドファイルに記録している。検索時には、検索条件語を構成する各文字の位置情報をインバーテッドファイルから読み出し、その位置関係をチェックすることで検索条件語の出現を判定する。

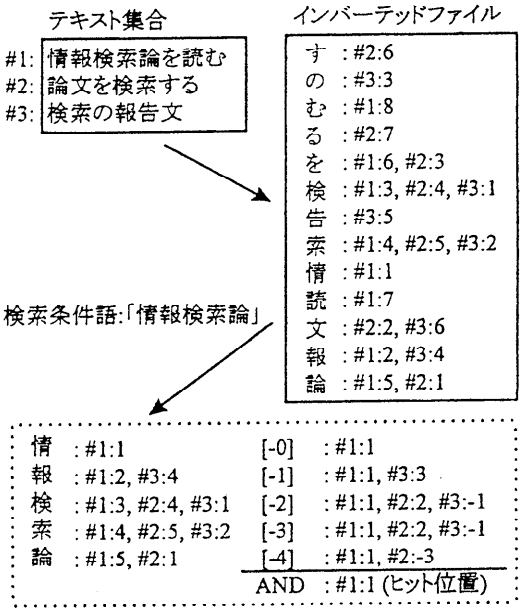

図1: 文字列インバージョンのナイーブな実装例

このような実装において検索レスポンスを大きく左右するのは、インバーテッドファイルのキー部分の検索時間よりも、むしろ位置情報データの読み出し時間である。テキスト中に頻出する文字が検索条件語に含まれた場合、その文字の位置情報データを読み出すための時間が、検索レスポンス時間の大半を占める。その対策としてキー文字列長を長くすれば、検索条件語をキー文字列に分割する際に、その分割数を少なくできる(1文字ごとに位置情報データを読み出さなくてよい)。さらに、キーが細かく場合分けされることになるので、対応する位置情報データも場合分けに応じて絞り込まれる[2]。

筆者らはこの点に着眼し、位置情報データの読み出し量を削減することで検索レスポンスを高速化する方針をとり、FSI法を設計した。

3 インバーテッドファイルの構造

FSI法では、上記の設計方針に基づき、(1)キー文字列長の字種別可変化、(2)縮退文脈の付与、(3)位置情報データの圧縮などを導入した。また、一般にインバーテッドファイルの容量と検索レスポンスはトレードオフ関係になる。このトレードオフは、要因をパラメータ化し、検索対象テキストの文字列統計からそのパラ

Flexible String Inversion Method for High-Speed Full-Text Search
Toshikazu Fukushima and Susumu Akamine
NEC Corporation

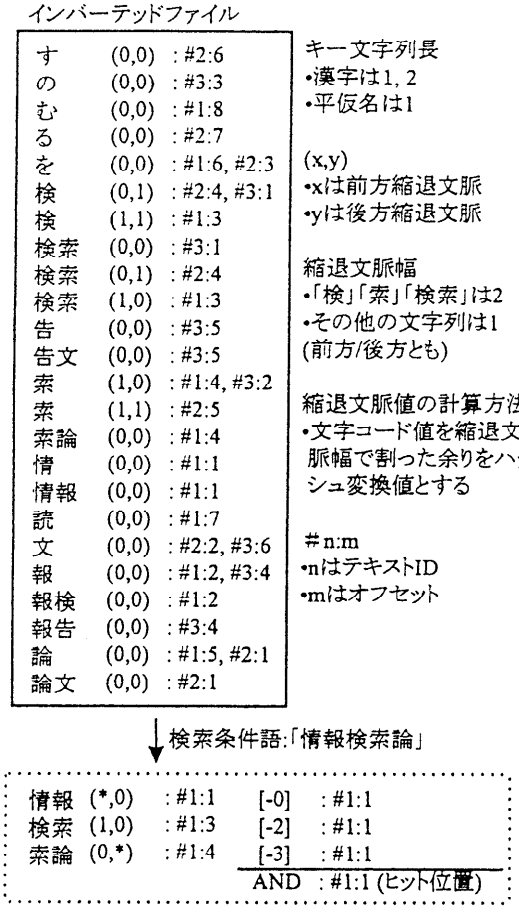

図 2: FSI法のインバーテッドファイルと検索例

メータ値を決定することで軽減できる。アプリケーションに応じた柔軟あるいは最適な調整も可能になる。

3.1 キー文字列長の字種別可変化

前述したように、キー文字列長を長くすることで検索時の位置情報データの読み出し量を削減できる。しかし、キー文字列長に対してキーの種類は累乗オーダーで急増するので、一律に長くすることは得策でない。高頻度文字(対応する位置情報データ量が多い)に対してキー文字列長を長くするのが容量面の効率がよい。

FSI法では、字種(漢字/平仮名/片仮名/英字/ほか)ごとに分けて、キー文字列長を設定するようにした(パラメータとして変更可能)。図2の例では、キー文字列長を漢字は1・2、平仮名は1としている(オリジナルテキストは図1と同様)。例えば、テキスト#2「論文を検索する」からは「論」「文」「を」「検」「索」「す」「る」「論文」「検索」がキー文字列として取り出されている。

3.2 縮退文脈の付与

前節の方策では調整のしやすさも考慮して字種別にパラメータを設定するようにしたが、同じ字種の文字列でも出現頻度にはかなりばらつきがある。そこで、縮退文脈の付与によって、さらにキーを細分化し、位置情報データを絞り込めるようにした。

ここでいう縮退文脈とは、キー文字列の前後の各1文字の文字コードを、ある値(=縮退文脈幅)未満の値にハッシュ変換したものである。その際の縮退文脈幅は、各キー文字列の出現頻度に基づいて個別に設定できる。すなわち、高頻度のキー文字列については縮退文脈幅を大きくすることで細かく分類し、低頻度のキー文字列については縮退文脈幅を1(=縮退文脈なしと同じ)とするようなコントロールが可能である。

図2の例では、出現頻度の高い「検」「索」「検索」について、縮退文脈(縮退文脈幅は2)を付与することで、位置情報データを振り分けている。縮退文脈幅が2ということは、前後の文字を0または1にハッシュするという意味で、(#1:3)の「検」を例とすれば、直前(#1:2)の「報」→1、直後(#1:4)の「索」→1というハッシュ変換の結果、縮退文脈の値が(1,1)となっている。

3.3 位置情報データの圧縮

位置情報データの読み出し量を削減するには、データ圧縮をかける手もある。1要素(テキストID+オフセット)に固定ビット数を割り当てた配列の形式が、位置情報データの最もナイーブな表現形態であるが、FSI法では、直前の要素値との差分をとった上で、NULLバイトを削除して可変長化し、要素の区切りにフラグビットを付与する方法で圧縮している。

4 検索手順

検索条件語が与えられたら、まず、その文字列を字種別に分割し、次に、各字種の文字列を、登録時に字種別に定めたキー文字列長にしたがい、なるべく少ない分割数で、なるべく長いキー文字列を取り出す。さらに、検索条件語における各キー文字列の縮退文脈を求め(ただし検索条件語の先頭と末尾の縮退文脈は任意値とする)、それを付与したキー文字列でインバーテッドファイルを検索する。その結果として得られた位置情報について従来と同様に位置関係をチェックし、検索条件語の出現を判定する(図2参照)。

5 おわりに

高速全文検索のためにFSI法を提案し、その方式概要を述べた。FSI法では、位置情報データの読み出し量を削減する3つの方策を導入し、ファイル容量と検索速度のトレードオフに対して文字列統計に基づく最適あるいは柔軟な調整を可能にした。FSI法の実装と評価結果に関しては別論文[3]にて報告する。

参考文献

[1] 多田ほか、文字成分表を用いた大規模全文検索方式の開発ーハッシュレス文字成分表の高精度化方式ー、情処51全大:7E-6、1995年。
[2] 菊池、日本語文書用高速全文検索の一手法、信学論:J75-D-I(9)、1992年。
[3] 赤峯ほか、高速全文検索のためのフレキシブル文字列インバージョン法(2)実装と評価、情処53全大:5T-5、1996年。