1C-6

類似文字列検索におけるLCP配列を用いた索引の提案

木村 光樹 † 安達 淳 †† 高須 淳宏 ††
† 東京大学大学院 †† 国立情報学研究所

1 はじめに

類似文字列検索は，単純な spell check や web 検索でのクエリ修正，データベース中のデータクリーニングや record linkage などと実世界での応用範囲も広い．類似文字列検索はクエリと類似度がある閾値以上のデータセット中の文字列を全て列挙することと定義される．しかし，類似度の計算コストが大きく，クエリをデータセット中のデータ全てとの類似度を計算することは現実的ではない．そのため類似度の計算回数を減らすために，類似度の高い文字列同士は gram と呼ばれる部分文字列が多いことに着目して枝刈りを行う，gram ベースの索引付けを用いた手法の研究が盛んに行われている．gram ベースの索引付けを用いた類似文字列検索では，一般的に gram 長，N は固定して用いられることが多く，N の値によって次に示す特徴が現れる．N が大きいと文字列のリスト内に含まれる文字列数が少なくなるが，索引サイズが大きくなる．逆に N が小さいと索引サイズが小さくなるが，文字列のリスト内に含まれる文字列数が多くなる．このため N の値が性能に大きく影響することがある．そこで，近年では gram 長を固定せずに用いる可変長 N-gram を用いる研究も行われている．可変長 N-gram を用いることで gram 長に関する問題に柔軟に対応できるようになってきたが，索引語の抽出および索引語辞書の保持を多分木により実現しているため，索引構築時の時間計算量，空間計算量が大きいという問題がある．そこで，本論文では時間計算量，空間計算量を改善するために線形時間で構築が可能な LCP 配列を用いた索引語抽出の手法を提案する．本手法を適用した予備実験では，既存の手法に比べて索引語抽出にかかる時間が大幅に削減されることが確認された．

2 提案手法

本論文では，LCP 配列を用いた可変長 N-gram とそれを用いた索引付けの手法を提案する．

2.1 表記

T をテキスト，$|T|$ をテキスト長，$T[i]$ を T の $i(1 \leq i \leq |T|)$ 番目の文字，$T[i:j](1 \leq i < j \leq |T|)$ を $T[i]$ から始まり $T[j]$ で終わる T の部分文字列とし，T_i を T の $T[i]$ から始まる接尾辞とする．テキスト T の接尾辞配列を SA_T で表し，$lcp(T,S)$ を文字列 T と S の共通接頭辞の最長長とする．

2.2 LCP 配列を用いた索引語の抽出

可変長 N-gram は，固定長 N-gram の大小の利点をうまく利用するために考案された．索引語のサイズの肥大化を防ぐために，索引語として用いる長い gram はデータセット中に出現する頻度の多いものである．可変長 N-gram の先行研究である Li らの提案した VGRAM[2] では，gram の最長長 N_{max}，最短長 N_{min}，頻度の閾値 τ の 3 つのパラメータを与え，データセット中に出現する部分文字列のうち長さが N_{min} 以上 N_{max} 以下で出現頻度が τ より大きい部分文字列を基準に可変長 N-gram の抽出を行っていた．しかし，VGRAM では木構造を用いて可変長 N-gram の抽出と索引付けを行っているために，大きなコストを要してしまう．さらに，VGRAM を用いることで，クエリを処理する時間は固定長 N-gram を用いるときと比較し，性能が向上したことが報告されていたが，パラメータが 3 つ必要であるためにチューニングのコストが大きく，パラメータを変える度に木を作り直さなければならないのが大きな問題である．

そこで筆者らは時間，空間効率を高めるために LCP 配列と呼ばれる配列を利用した索引語となる可変長 N-gram の抽出手法を提案する．LCP 配列は，接尾辞配列上で隣り合う 2 つの文字列の最長一致する共通部分接頭辞長を集めた配列であり，$LCP[i] = lcp(T_i, T_{i+1})(1 \leq i < |T|)$ を満たす．接尾辞配列は線形時間で構築できることが知られており，接尾辞配列が求まっていれば LCP 配列も線形時間で構築できることが知られている [1].

接尾辞配列上で任意の 2 地点 $i, j(1 \leq i < j \leq |T|)$ での lcp の値は次式を満たすことが知られている．

$$lcp(T_i, T_j) = \min_{i \leq k < j}(LCP[k]) \quad (1)$$

LCP 配列と接尾辞配列を用いて，データ中である出現頻度 (τ) 以上の長さが n 以上の部分文字列を全て

LCP Array Indexing for Approximate String Matching

† Mitsuki Kimura(mick@nii.ac.jp)
†† Jun Adachi(adachi@nii.ac.jp)
†† Atsuhiro Takasu(takasu@nii.ac.jp)
The University of Tokyo (†)
National Institute of Informatics (††)
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo,101-8430 Japan

Copyright ©2012 Information Processing Society of Japan.
All Rights Reserved.

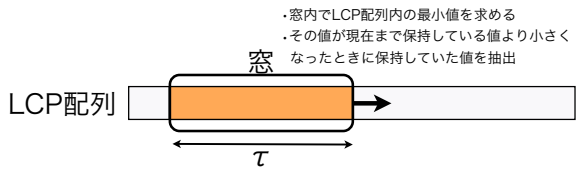

図 1: LCP 配列を用いた可変長 N-gram 抽出

抽出することを考える．まずデータセット中の全ての文字列をデータ中に出現しない文字を使って連結し，一つの長い文字列をつくる．次に，その出来上がった文字列の接尾辞配列を求める．そして，求めた接尾辞配列からその文字列の LCP 配列を求める．接尾辞配列では接尾辞を辞書順に並べたもののテキスト中での出現位置の配列である．このため，あるテキスト中の部分文字列 s_{sub} を接頭辞としてもつ接尾辞が出現する位置は配列上で固まって現れる．今 s_{sub} を接頭辞としてもつ接尾辞のうち辞書順で最も小さいものが配列上で i の位置に位置し，最も大きいものが $i+l$ に位置していたとすると，このテキストに s_{sub} が出現する頻度は $i+l-i+1=l+1$ である．このことと式 (1) により，出現頻度が τ 以上で長さが n 以上の部分文字列を全て見つけるためには，LCP 配列に対して長さ τ の窓を配列上全て動かし，その窓内で最も小さい配列の値が n を超えていれば実現できる (図 1)．このとき，ある可変長 N-gram の接頭辞となるような可変長 N-gram を抽出しないようにするために，抽出は前回の窓で見つかったものより小さい値が抽出されたとき，前回の窓内で見つかったものを抽出する．

2.3 索引付け

可変長 N-gram を用いて索引付けを行うときには，固定長のときと違い複数パターンの索引付けが可能となる場合がある．そのため，転置索引中の各レコードに含まれる文字列数を減らすために，文字列中で最長一致する可変長 N-gram を索引語とし，また他の索引語の部分文字列となる可変長 N-gram を索引語としない．このことを踏まえて，索引付けを行う．まず抽出した可変長 N-gram を抽出したときと同様に一つの文字列 D^g にし，その接尾辞配列,SA_D と LCP 配列,LCP_D を求める．今索引付けしたい文字列を s とすると，次に示すようにして文字列 s 中の索引語を出す．これは $i=1,k=0$ を初期値として与え，i を 1 ずつ増やしながら，$i+l=|s|$ を満たすまで繰り返す．

1. $D^g_{SA_D[j]} \leq s[i:|s|] < D^g_{SA_D[j+1]}$ を満たす j を見つける
2. LCP_D を用いて $l=lcp(s[i:|s|], D^g_{SA_D[i]})$ とする
3. $i+l>k$ を満たすときは，$k=i+l$ とし，$s[i:i+l]$ を索引語とする

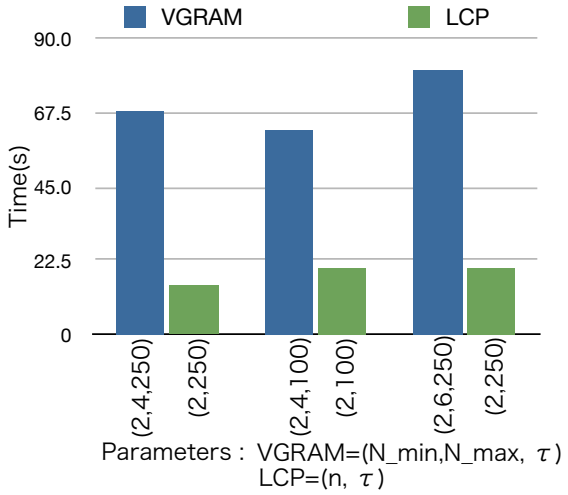

図 2: 索引語抽出時間の比較

3 評価実験

実験ではエントリー数が 69,069 の英単語辞書を用いて実験を行った．単語長は平均 9.4 文字であった．比較に用いた手法は VGRAM であり，比較した内容は可変長 N-gram を抽出するのに要した時間と，抽出した可変長 N-gram を用いてデータセット中の索引語全てを索引付けしたのに要した時間である．VGRAM，提案手法ともにパラメータを変えて実験を行った．結果を図 1 に示す．図から分かるように索引語となる可変長 N-gram の抽出は，提案手法が 3 倍程度速い．また，索引付けでは VGRAM の各パラメータを $(N_{min}, N_{max}, \tau) = (2,4,250)$，提案手法では $(n,\tau)=(2,250)$ を用いて比較した．それぞれ要した時間は 12.7(s) と 11.4(s) であり，提案手法が長い gram が抽出されているにも関わらず，同程度の時間で済むことが確認された．

4 おわりに

gram ベースの索引付けを用いた類似文字列検索において，可変長 N-gram の抽出に木構造を用いていた既存の手法の抽出コストが大きくなる問題に対して，LCP 配列を用いて抽出のコストを抑える手法を提案し，予備実験においてその性能が証明された．今後は，データ中に出現する文字種が多い日本語等への対応を考えていきたい．

参考文献

[1] Landau et al. Linear-time longest-common-prefix computation in suffix arrays and its applications. In *Springer LNCS*, 2006.

[2] Li et al. Vgram: improving performance of approximate queries on string collections using variable-length grams. In *VLDB*, 2007.

Copyright ©2012 Information Processing Society of Japan.
All Rights Reserved.