D-028

高頻度語を可変長索引語に用いる類似文字列検索手法の検討

Variable Length Index Using Frequent Patterns for Approximate String Search

木村 光樹 † 高須 淳宏 ‡ 安達 淳 ‡
Mitsuki Kimura Atsuhiro Takasu Jun Adachi

1 はじめに

近年ビッグデータの出現により，類似文字列検索の需要は大きくなってきている．大規模データからマイニングを行い，よい解析結果を得るためには対象となるデータの品質を保証する必要があり，類似文字列検索では，このデータの品質の保証をするためには大きな力を発揮する．また，それ以外にも単純なスペルチェックに用いられたり，web 検索におけるクエリ修正等に用いられるなど，類似文字列検索は現代の情報分野での基盤となるアプリケーションとして重要な位置を占めている．そのため，近年の情報量の増加とともにその高速さと正確性が，より求められるようになってきた．

文字列間の類似度を表す指標は，編集距離，Jaccard Similarity, Cosine Similarity, Dice Similarity など様々に存在する．しかしながら，これらの文字列間の類似度を計算するのに一般的に時間がかかってしまうことが知られているために事前に解候補の削減を行う必要がある．一般的な類似文字列検索では，類似した文字列同士は共有する部分文字列が多くなるということに着目し，解候補の削減を行う．そのため，類似文字列検索における索引付けでは，索引語の単位をある部分文字列とすることがほとんどである．この索引語長は一般的に固定して用いられることが多いが，索引語長が短いときには，索引サイズは小さくなる一方，短い部分文字列を共有する部分文字列は多いため，解候補の決定に時間がかかることが知られている．また，逆に索引語長を長くすると，索引サイズが大きくなってしまう一方で，長い部分文字列を共有する部分文字列は少ないため，解候補の決定が容易に行えるという利点が存在する．このため，索引語長によって類似文字列検索システムの性能が大きく変わってしまうという問題が生じてしまう．このことを踏まえて，索引語長の長いとき，短いときの両方の利点のみをうまく使えるようにするために，近年では索引語長を可変にするという研究もなされている [2]．可変長索引語の先行研究である VGRAM[1] では，木構造を用い，データセット中で出現する部分文字列の頻度を考慮して，可変長索引語を決定し，検索に用いるという手法を提案している．VGRAM では，固定長の索引語と比較して，検索にかかる時間を改善しており，この索引語が種々の検索手法に適用できるという点において，優れた手法となっている．しかしながら，索引語を決定するのに必要なパラメータが 3 つ存在しており，あるデータセットでの最適なパラメータを求めるのに非常に大きい労力がかかってしまう．

このことを踏まえて筆者らは，接尾辞配列と LCP 配列を用いて，データセット中に出現する頻度が大きい部分文字列を索引語とする手法を提案する．本手法では，索引語の決定に必要なパラメータは 2 つであり，パラメータを決定したときに，データセット中から索引語を求めることは，線形時間で可能である．そのため，VGRAM と比較してあるデータセット中で最適な索引語を決定し，索引付けを行うのに要する時間を大幅に削減することができた．また，この索引語を用いて，実際に検索を行ったときにかかる時間は，VGRAM とほぼ同等であったことを実験で示すことができた．

† 東京大学大学院

‡ 国立情報学研究所

2 問題定義

本稿では，数ある類似度のうち最もよく使われる，編集距離を対象とする．編集距離はある文字列を別の文字列へと変換するために行う編集操作の最小回数で表される．編集操作とは，挿入・削除．置換の 3 つの操作のことであり，それぞれの操作を行うごとに距離が 1 ずつ増えていく．

文字列 s_1,s_2 間の編集距離を $ed(s_1, s_2)$ で表す．クエリを Q，文字列データの集合を S としたとき本稿で対象とする類似文字列検索とは，ある閾値 k を用いて $ed(s, Q) \leq k$ を満たす $s \in S$ となる文字列 s を全て見つけることである．

3 提案手法

提案する高頻度索引語抽出手法とそれを用いた索引付け手法について紹介する．

索引語抽出手法

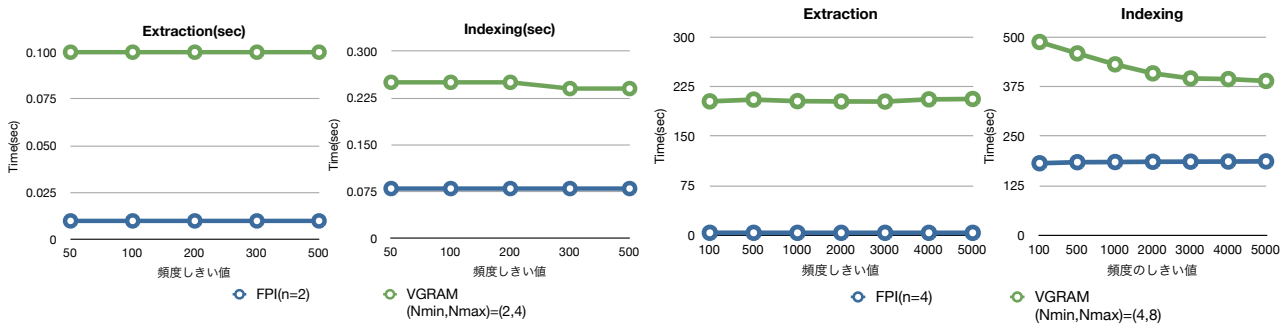
索引語抽出には，接尾辞配列と，接尾辞配列上で隣り合う部分文字列の共通すつ接頭辞の最長長を収めた LCP 配列を用いる．接尾辞配列は，線形時間で構築する手法が既に提案されており，接尾辞配列が既知であれば，LCP 配列も線形時間で構築する手法が提案されている．

本提案手法における，高頻度語とは，長さが n 以上でかつ，データセット中で出現する頻度が τ 以上である部分文字列であると定義する．以下にその抽出アルゴリズムの概要を示す．

1. データセット中の全ての文字列を，そのデータセットに出現しない特殊文字を用いて連結し，一つの文字列とし，この接尾辞配列と LCP 配列を求める．
2. LCP 配列の値を先頭から見ていき，各配列値の最大の配列位置とその値を記憶しておき，配列の走査個数 cnt が τ 個になるまで走査した後，記憶している LCP の値の最小値 $= can.lcp$ とその配列位置 $= can.pos$ を候補とする．
3. $can.pos$ となっている配列値を全て消去し，$cnt = \tau - can.pos$ とし，走査を再開する．
4. 次に，$cnt = tau$ となったときに，前回の $can.lcp$ 今回の $can.lcp$ が大きくなっていれば，$= can.lcp$ と $= can.pos$ を更新する．
5. 抽出は走査中に，$can.pos$ より小さい値が見つかったときに $= can.lcp$ と $= can.pos$ を高頻度語として抽出する．

このアルゴリズムでは，最初と LCP 配列の値が n より小さい値が見つかったときには抽出の基準となる値を n に書き換え，cnt を 0 に設定する．

本アルゴリズムの特徴として，データセット中の接尾辞配列と LCP 配列を一度求めておけば，パラメータを変更したとしてもこの配列を作り直す必要がなく，抽出アルゴリズムを用いるだけでよいことが挙げられる．



(a) Dict　　(b) DBLP

図 1

索引付け手法

索引語抽出アルゴリズムを用いて，抽出した索引語でデータセット中の索引語を索引付けする手法について紹介する．まず，抽出のときと同様に，索引語を一つの文字列として，その文字列の接尾辞配列と LCP 配列を求める．本手法では，可変長の索引語を用いるために，一つの文字列に大して一意に索引語が決まらないことが生じてしまうため，以下の制約を課す．

1. 索引語と最長一致するものを索引語とする．
2. 他の索引語の部分文字列である索引語は索引語としない．

以上二つの制約を考慮して，実際に索引付けする手法について紹介する．索引語を決定するためには，接尾辞配列と LCP 配列，接尾辞配列の rank 配列を用いる．接尾辞配列の rank 配列は，接尾辞配列を SA，rank 配列を R としたとき，$R[SA[k]] = k$ の性質を満たす配列のことである．rank 配列は，接尾辞配列の逆関数のようなものであり，その定義から簡単に線形時間で求めることができる．この配列を使えば，例えばある部分文字列 p が接尾辞配列上で出現する箇所が pos という位置だったとすると，部分文字列 p の 2 文字目から始まる部分文字列の出現箇所は，$R[SA[pos] + 1]$ として求めることができる．

索引語決定の概略は以下のようになる．

1. 対象となる文字列 s の先頭から見ていき索引語集合内で最長一致する索引語をこの文字列の索引語とし，この文字列の索引語集合 $VG(s)$ に追加する．このとき，この索引語長を m とし，その索引語の接尾辞配列上での出現位置を pos とする．
2. $R[SA[pos] + 1] + m - 1$ の付近で LCP 配列の値が $m - 1$ となる範囲で文字比較を行い，最長一致するものを索引語候補とし，その索引語長を m とし，接尾辞配列上での出現位置を pos とする．
3. この索引語候補が決定した索引語の部分文字列であれば，その索引語は破棄し，部分文字列でなければ，$VG(s)$ に追加する．
4. $VG(s)$ に含まれる索引語の終端が文字列 s の終端に一致するまで $2 - 3$ を繰り返す．

4　評価実験

実験では比較手法として，VGRAM を用いた．比較した内容は，索引語抽出にかかる時間と索引付けに要した時間，検索に要した時間の 3 つである．実験に使用したデータは表 1 の通りである．

表 1　実験で使用したデータセット

Data	Size	平均長	文字種	説明
Dict.	635K	9.4	26	English dictionary
DBLP	78M	104.5	93	bibliography data

表 2　配列構築時間

	接尾辞配列 (sec)	LCP 配列 (sec)
Dict.(sec)	0.07	0.04
DBLP(sec)	75.92	24.62

提案する手法では，事前に接尾辞配列と LCP 配列を求める必要があり，これらの配列を求めるのに要した時間を表 2 に示す．実験の結果を図 1 に示す．Dict,DBLP のどちらのデータの場合でも，索引語抽出と索引付けに要する時間が大幅に削減されたことが分かる．また，検索に要した時間は，データセット中からランダムに 10,000 個の文字列を選択し，編集距離を 1 としたものをクエリとしたときに，Dict の場合，提案手法 (FPI) では 23.05(sec)VGRAM では 27.71(sec) であった．

5　おわりに

接尾辞配列と LCP 配列を用いた高頻度語抽出手法を提案し，それを索引語に用いる類似文字列検索システムについて紹介した．既存の可変長索引語手法において，検索にかかる時間はそのままで，索引語抽出に要する時間と索引付けにかかる時間が大幅に削減されたことが実験で示すことができた．本手法は，既存手法に比べてパラメータの数も減っており，データに依存する最適パラメータを求める点で優位であると言える．

参考文献

[1] Li, et al., VGRAM: Improving Performance of Approximate Queries on String Collections Using Variable-Length Grams. In VLDB, 2007.

[2] Jianbin Qin et al.,Efficient Exact Edit Similarity Query Processing with Asymmetric Signature Schemes. In SIGMOD, 2011.