6M-7

LSPC を用いた活字文書画像からの頻出文字列抽出

Frequent string mining from document images using LSPC

細谷拓史　　寺沢憲吾
Takumi Hosoya　　Kengo Terasawa

公立はこだて未来大学
Future University Hakodate

1. はじめに

本研究では文字認識されていない文書画像から頻出文字列の抽出を行うことを目的とする．

近年インターネット，デジタルアーカイブの普及により，多くの文献史料がデジタル画像化され公開されている．このような多くの文献史料を扱う場合，インデックスの作成，文書内容の推定に頻出文字列を利用することが有効である．しかし，文献史料は書体，用法など現在とは異なるものが画像形式で保存されているため，画像からの OCR による文字認識は困難である．その中で函館市中央図書館，デジタル資料館にて公開されている「文書画像システム」では，文書画像から検索対象とした文字列画像と似ている文字列画像を探し出すことができる．

このシステムを使った Terasawa らの研究で，近似近傍探索手法の 1 つである LSH(Locality-Sensitive Hashing)[1]の考えから LSPC(Locality-Sensitive Pseudo-Code)[2]を開発し，高速に画像による全文検索を行えるようになっている．

しかし，現在のシステムの検索では頻出文字列を抽出することに対応していない．そこで本研究では，文書画像検索システムでの頻出文字列の抽出を可能にすることを目的とした．また使用する画像は，文書画像検索システムと同様に函館市中央図書館，デジタル資料館にて公開されている 1881 年の「函館新聞」(図 1)である．

図 1　函館新聞の一部

2. 文書画像検索システム

文書画像検索システムは，文書画像と画像に対応した特徴量ベクトルを使用して文字列検索を行う．特徴量ベクトルは特徴量が保存されているが，今回使用する函館新聞では，画像内の 1 文字毎の「文字領域」の特徴量を保存している．つまり，画像内に 2000 文字含まれていたら，2000 文字個々の特徴量を保存している．

特徴量ベクトルは 128 次元，各成分は非負の実数，ノルムが 1 となるように正規化されている．検索においては，検索したい語句部分の画像「クエリ画像」の特徴量と他の特徴量の距離計算を行う．その結果は 0 から√2 までの実数になり，0 に近いほどクエリ画像と似ていることを示す．

LSPC[2]では，検索の高速化のため，特徴量ベクトルを擬似コードに変換して検索を行う．擬似コードは特徴量ベクトルを基に作られる整数値の組である．擬似コードを使用した検索では，クエリ画像の擬似コードと他の擬似コードを比較し，整数の組のうち少なくとも 1 つが一致(semiequivalent)した時に似ている画像と判別する．この方法は特徴量ベクトルによる検索に比べ，精度は若干落ちるが，実数の計算を行わずに整数の比較だけで済むために，計算時間が大幅に短くなる．

3. 実験

実験には，文書画像検索システム，函館新聞の 1881 年 10 月 22 日の 1 ページ目の画像を使用する．この画像には特徴量を保存した文字領域が 2017 個含まれる．

頻出文字列を調べるためには，検索システムの出力を使い，同じ文字列と違う文字列の識別を行う必要がある．検索システムでは 2 種類の検索方法が存在するが，擬似コードは特徴量ベクトルを基礎として作られるものであるので，まず特徴量ベクトルを使った段階で同じ文字列を調べる．特徴量ベクトルを使った特徴量の計算は，2 特徴量間の距離が 0 に近いほど，似ている文字であるため，同じ文字と違う文字を識別する閾値が存在すると考えた．次いで，特徴量ベクトルで閾値を設定した後，その閾値を用いて擬似コードを作成し大規模データに対応させる．

実験は以下の段階で行う．まず，特徴量ベク

トルの段階で閾値の設定を行い，次に閾値以下の距離となる文字列の組み合わせをすべて抽出できるように擬似コードを設計する．得られた擬似コードを用いて，抽出した組み合わせをクラスタリングしグラフ化する．作成したグラフから頻出文字列となるものを決定する．

最終的には任意の文字数の頻出文字列抽出を目標としているが，今回は 3 文字列の組み合わせの抽出までを行う．

3.1 実験 1

まず，閾値を設定するために，特徴量ベクトルにて，画像内に複数存在しており，文字の複雑さの違いから選んだ 3 つの文字領域，「ノ」「明」「第」と文書画像内の全ての文字領域と距離計算を行い，ヒストグラムを作成する．また，「ノ」「明」「第」のそれぞれと同じ文字との距離計算も行い，各階級にて同じ文字の割合を求める．この時ヒストグラムの階級は√2/10 間隔とする．

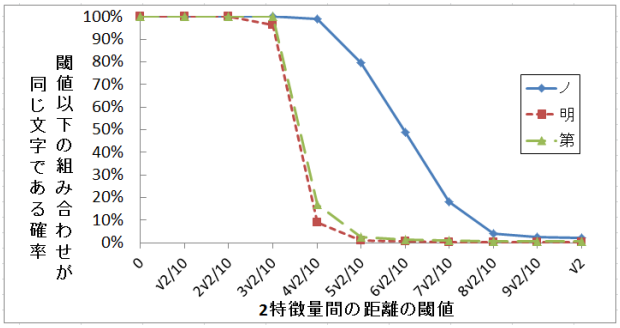

図 2 同じ文字を検出する確率

結果を図 2 に示す。「ノ」「明」「第」いずれにおいても同一文字の割合が，距離 3√2/10 までは高いが，それ以上では急落する．これにより，同じ文字を検出する閾値は 3√2/10 に設定することが有効であるとわかる．

実際に，同様の実験環境で 3 文字が連続して 3√2/10 以下となるものをすべて抽出した結果，そのような組み合わせは 88 組存在し，それらは全て同じ文字列の組み合わせであった．

3.2 実験 2

実験 1 で設定した閾値を用いて，同一の 3 文字列を 90%以上の確率で検出する擬似コードを作成する．擬似コードには精度と計算速度を支配するパラメタ k が存在するが，今回は k=4, k=6 の 2 種類について実験を行う(パラメタ L は k から自動的に定まる)．その後擬似コードにて 3 文字列の抽出を行う．その結果に実験 1 で行った 3 文字列での 88 組が 90%以上含まれることを確認する．

表 1 k=4, k=6 の時の擬似コードを用いた 3 文字列検索結果

k	L	同じと判断した文字列の組	特徴量ベクトルでの検索結果をどれだけ含むか	検索時間
4	78	918	88	0.146
6	382	192	88	0.851

結果を表 1 に示す． k=4, k=6 共に実験 1 での 88 組が全て含まれており，擬似コードで同一の 3 文字列の抽出ができることが確認された．検索時間は 0.14~0.85 秒であり，特徴量ベクトルによる検索時間(約 7 秒)に比べ大幅に高速化されている．一方，LSPC は確率的アルゴリズムであるため，同一でない 3 文字列も対応として検出してしまう．そのため，クラスタリングでこれから真の対応だけに絞り込む．

4. 結論と今後の予定

今回の実験にて，擬似コードを使用し頻出文字列の候補となる文字列の組み合わせを抽出することができた．しかし，この候補は違う文字列も多く含むため，同じ文字列として真に対応しているものに絞り込む必要がある．絞り込むための手法としてクラスタリングによるグラフ化(図 3)を考えている．

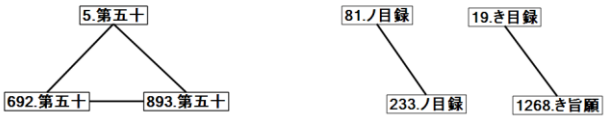

図 3 グラフ化を行った時の例

グラフの中から密な連結成分(図 3 左側)だけを採用し，小さな連結成分(図 3 右側)を棄却すれば，真の対応だけが残るのではないかと予想される．今後の予定は，LSPC にて函館新聞 1 年分の画像に対しこの手法を適用し，頻出文字列が抽出できるかどうかの確認を行う．

参考文献

[1] G. Shakhanarevich, T. Darrell and P. Indyk, “Nearest-Neighbor Methods in Learning and Vision (Theory and Practice),” The MIT press, Cambridge, MA, 2005.

[2] K. Terasawa and Y. Tanaka, “Locality Sensitive Pseudo-Code for Document Images,” Proc. ICDAR2007, vol. 1, pp. 73-77, 2007.