D-028

文書に特徴的な単語を考慮した検索結果のクラスタリング
A Clustering Method for Retrieval Results with Feature of Words in documents

長尾 壮史 † **山岸 英貴** † **平澤 茂一** ‡
Masafumi Nagao Hidetaka Yamagishi Shigeichi Hirasawa

1. はじめに

近年，インターネットの急速な普及によって Web ページが急増している．インターネットによる情報検索においてサーチエンジンが最も利用されている手段である．サーチエンジンの検索結果が膨大になるとその検索結果はいくつかの分野 (トピック) に分かれると考えられる．そのためランキング上位の検索結果が必ずしもユーザの意図する情報とは限らないという問題が生じる．

この問題に対処するため，サーチエンジンによる検索結果のリストをクラスタリングし，同一トピックを持つ集合に分類することで，目的ページを素早く発見するという手法がある [1].

本研究では文書を特徴付けるフレーズとその概念を考慮したクラスタリング手法を提案する．また，本手法を小規模な Web ページ集合を対象にして実験を行い，その有効性を示す.

2. 従来手法

これまで数多くの文書クラスタリングアルゴリズムが提案されているが，Suffix Tree Clustering アルゴリズム [2] は検索結果のクラスタリングに適したアルゴリズムとして知られている.

2.1 Suffix Tree Clustering

Suffix Tree Clustering アルゴリズムは，サーチエンジンから得られた検索結果を対象データとし，1 件の検索結果を 1 つの文書とみなす.

[Suffix Tree Clustering アルゴリズム]

s1) Suffix Tree を用いて文書間に共通するフレーズを抽出する.

s2) 共通フレーズにスコアを付け，閾値以上のものをすべて抽出する．抽出されたフレーズを含む文書集合を初期のクラスタとする.

s3) 2 つのクラスタをくり返し連結し，最終的なクラスタを出力する. □

次にアルゴリズムの詳細について説明する.

2.1.1 Suffix Tree による共通フレーズ抽出

Suffix Tree は文の後接語を木構造に展開したものである．例えば，“cat ate cheese ”の 1 番目の後接語は “cat ate cheese ”，2 番目は 1 単語落とした “ate cheese ”，3 番目は 2 単語落とした “cheese ”である．これらを木構造で展開すると Suffix Tree となる．Suffix Tree を用いると，文書間に共通するフレーズを高速に発見できる．図 1 に文書 1 を “cat ate cheese ”，文書 2 を “mouse ate cheese ”として Suffix Tree に展開したものを示す．ここで葉のラベルは文書の番号と，何番目の後接語であるかを表す．例えば，図 1 のラベル (1,2) は，“ate cheese ”が文書 1 の 2 番目の後接語であることを表している．共通したノードに印をつけておき，最終的にその印のついたノードが共通するフレーズということになる．図 1 では “ate cheese ”と “cheese ”が文書間に共通するフレーズなので，それらのノードに a,b という印がついている.

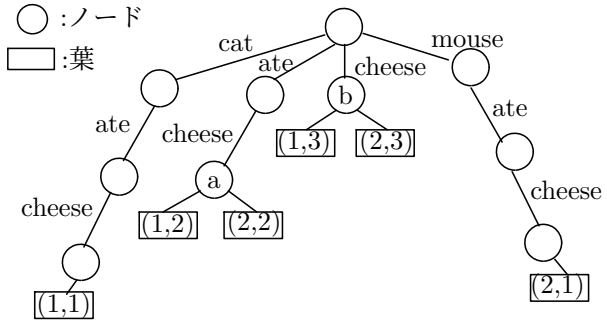

図 1: Suffix Tree の例

2.2 初期クラスタの生成

Suffix Tree によって抽出された共通フレーズを以下の (1) 式よってスコア付けする.

$$score(ph) = B(ph) \times len(ph). \quad (1)$$

$B(ph)$：フレーズ ph を含む文書数
$len(ph)$：フレーズ ph の単語数

文書数が多く，単語数が長ければ良いフレーズであるとし，スコア付けを行う．スコアが閾値を満たすフレーズをすべて抽出し，このフレーズを含む文書集合が初期クラスタとなる.

2.3 初期クラスタの連結

2 つの初期クラスタの共有文書が両者のクラスタに占める割合を調べるため，すべての組み合わせについて (2) 式を計算する.

$$\frac{|B_x \cap B_y|}{|B_x|} > \frac{1}{2} \quad \text{かつ} \quad \frac{|B_x \cap B_y|}{|B_y|} > \frac{1}{2} \ (x \neq y). \quad (2)$$

B_i：初期クラスタ i に含まれる文書集合
条件を満たすクラスタ同士を全て連結し，連結されたものが最終的なクラスタとなる.

2.4 従来手法の問題点

従来手法ではフレーズ選択のスコア付け方法が単純なため，初期クラスタはノイズ文書を多く含むものになりやすい．また結合方法も重複する文書が半分以上あれば結合するという粗い方法である．その結果，精度の低いクラスタを作る可能性がある．さらに，フレーズを基調としたクラスタリングのため，同じフレーズが表れなければ同じクラスタになることはないという問題がある．すなわち，言葉の多義性に弱いといえる.

3. 提案手法

クラスタ内のノイズ文書を減らすためには文書を特徴付けるフレーズの優先的な抽出が必要である．そのため新しいフレーズのスコア付け方法を与える．またそのフレーズが登場する文書をひとつのクラスタとする方法では語の多義性を考慮できないため，LSI（潜在的意味インデキシング）[3] を導入し，フレーズそのものではなく，フレーズの概念に基づいてクラスタを決定する.

†早稲田大学大学院理工学研究科経営システム工学専攻
‡早稲田大学理工学部経営システム工学科

3.1 相互情報量・タイトルによる特徴的フレーズ抽出

相互情報量は2つの情報源の関連度を表す指標である．これを用いて文書とフレーズの間の関連度を表す．文書との関連が高ければそのフレーズは文書を特徴付けているといえ，ノイズ文書を多く含む初期クラスタを減少させる．(3) 式が新たなスコア付け方法である．

$$mutualscore(ph) = \sum_{d_i \in D} p(ph, d_i) \cdot \log \frac{p(ph, d_i)}{p(ph) \cdot p(d_i)}. \quad (3)$$

d_i：全文書集合 D の各要素
$p(ph, d_i)$：文書 d_i においてフレーズ ph の出現確率
$p(ph)$：ph の出現確率　　$p(d_i)$：d_i の出現確率

検索結果は，ページのタイトルと抜粋文で構成されている．タイトルに出てくるフレーズはそのページを表す重要なフレーズであると考えられる．そこで提案手法では，スコアが高く，かつタイトルに出てくるフレーズを抽出フレーズとする．

3.2 概念によるクラスタの生成

従来手法は語を基調としたクラスタリング手法である．そのため，たとえば “automobile ”と “car ”という語は意味は同じであるが，処理上まったく別物であると判断され，片方の語が抽出フレーズとなっても，他の同義語を含む文書集合を抽出することができなかった．そこで語そのものではなく，語の概念を用いて文書集合を抽出する．語の概念を表現するために LSI を利用する．

3.2.1 LSI（Latent Semantic Indexing）

文書行列を $A = U\ \Sigma\ V^T \rightarrow A_k = U_k \Sigma_k V_k^T$ という形で特異値分解し，固有値の大きいものから k 次元を使うことにより次元圧縮する（U,V は左, 右特異値ベクトルと呼ばれる）．語－文書ベクトルから概念－文書ベクトルに圧縮することができ，文書の表現が語に依存しなくなる．そのため，その語が出現しない文書も抽出することが可能になる．

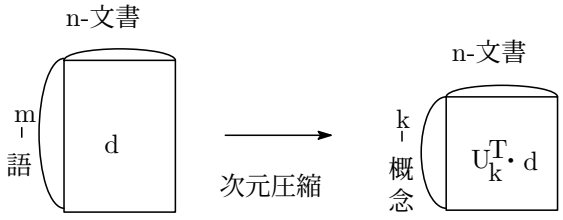

図 2: LSI の概念図

3.2.2 LSI を利用した概念による文書集合抽出

LSI によって圧縮された行列を用いて，抽出されたフレーズと文書間の類似度を式 (4) で計算する．抽出されたすべてのフレーズについて文書との類似度を計算し，類似度の高いフレーズのクラスタにそれぞれの文書が割り当てられる．

$$sim(\boldsymbol{d}, \boldsymbol{ph}) = \frac{(\boldsymbol{U_k^T} \cdot \boldsymbol{d}) \cdot (\boldsymbol{U_k^T} \cdot \boldsymbol{ph})}{|\boldsymbol{U_k^T} \cdot \boldsymbol{d}| \cdot |\boldsymbol{U_k^T} \cdot \boldsymbol{ph}|}. \quad (4)$$

$\boldsymbol{d}$：文書ベクトル
$\boldsymbol{ph}$：フレーズベクトル
$\boldsymbol{U_k}$：k 次元に圧縮された左特異値ベクトル

4. シミュレーションと考察

4.1 シミュレーション条件と評価方法

シミュレーションは Web サーチエンジンの Google に検索語を入力し，その検索結果を実験データとして利用した．Google カテゴリのラベルを正解トピックとし，検索語に対して返された検索結果 n =250 件（フレーズ数 m　1000）を用いて，適合率と再現率の二つを考慮した F 値という基準で評価を行った．

$$F\ 値 = \frac{2}{1\ /\ 適合率 + 1\ /\ 適合率}$$

4.2 結果

従来手法の Suffix Tree Clustering と提案手法を比較した実験結果（次元数 k =30）の一部を表 1 に示す．() 内はクラスタに含まれる正解文書数である．また図 3 に LSI の次元数を変化させたときの実験結果を示す．

表 1: トピックごとの平均 F 値の比較

トピック	従来手法	提案手法
Computer	0.579 (20)	0.597 (23)
Regional	0.305 (9)	0.361 (13)
Arts	0.391 (9)	0.400 (10)
Business	0.297 (7)	0.346 (14)
Home	0.852 (26)	0.866 (29)
Shopping	0.327 (9)	0.452 (14)
平均	0.458	0.504

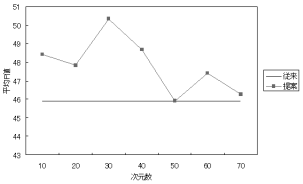

図 3: 次元数を変えたときの平均 F 値の比較

4.3 考察

1) 提案手法は従来手法より抽出された正解文書数が増加した．提案手法では LSI によりフレーズが出現する文書を抽出するではなく，フレーズの持つ概念が出現する文書を抽出してるため，ひとつのフレーズで従来手法より多くの正解文書を抽出できたと考えられる．

2) Home は F 値があまり向上しなかった．Home の正解文書の大部分は “Recipe ”という語を含んでいる．このように正解文書集合内に共通する一語がある場合は語をまとめて概念にする必要がなく，次元圧縮の効果が発揮できないと思われる．

3) 次元数が 10 〜 40 の低次元のときに，平均 F 値が従来手法より大きく向上した．これは低次元にすることで，うまく概念が形成されたと考えられる．また，高次元になるにつれて次元圧縮の効果が薄くなり，概念がフレーズそのものに近づいたため，従来手法とさほど変わらなくなっていると考えられる．

4) 今回は対象が小規模なデータであり，次元数がそれほど多くない．LSI というのは，非常に高い次元の文書ベクトルを圧縮し，精度の改善を図る手法である．したがってデータを増やし，次元を増やせば LSI がより効果的になり，クラスタの精度がさらに向上すると考えられる．

5. むすび

本研究では，相互情報量を用いて文書を特徴付けるフレーズを抽出し，次元を圧縮して得られたフレーズの概念を利用するクラスタリング手法を提案した．また小規模な Web のデータを用いた実験結果により従来より高い精度のクラスタリング結果を得られた．

今後は，重要フレーズ抽出のために検索語履歴や係り受け関係を利用した手法などを検討していきたい．

参考文献

[1] M. Hearst and J. Pedersen, “Reexamining the cluster hypothesis:Scatter/Gather on retrieval results, ”*Proc. 19th International ACM SIGIR Conference*, pp.76–84, 1996.

[2] O. Zamir and O. Etziomi, “Web document clustering: A fesirability demonstration, ”*Proc. 21st International AGM SIGIR Conference*, No.2 , pp.46–54, 1998.

[3] 北研二，津田和彦，獅々堀正幹，情報検索アルゴリズム，共立出版，2002.