1G2-03

少数フレーズに基づく文書クラスタリングとWeb検索への適用

Document Clustering with a small number of phrases and its application to the Web Search

岡部 正幸[*1] Masayuki Okabe
ビクター クリサノフ[*2] Victor Kryssanov
角所 考[*3] Koh Kakusho

豊橋技術科学大学[*1] Toyohashi University of Technology
科学技術振興機構[*2] Japan Science and Technology Agency
京都大学[*3] Kyoto University

This paper propose a document clustering algorithm which is an extended version of *Suffix Tree Clustering* algorithm. We improved two defects of STC, - phrase selection and similarity measure of cluster merging. Through the experiments we compared our method with several clustering algorithm. Then we indicate that our method outperforms not only the original STC but other well-known methods.

1. はじめに

Web 検索エンジンが返すヒットリストを加工し，目的ページをすばやく見つけ出すための工夫に関する研究が精力的に行われている．文書クラスタリングはその基盤となる技術であり，検索結果に適用できる高速かつ性能の高いアルゴリズムが必要となる．

これまで文書クラスタリングアルゴリズムは数多く提案されているが，Zamir らによって提案された *Suffix Tree Clustering* アルゴリズム（以下，STC）[Zamir 98] は，Web 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．このアルゴリズムは高速であり，またフレーズを利用しているため単語をベースにしたクラスタリング手法よりもクラスタの内容を分かりやすく表示できるという利点がある．一方，STC はクラスタ仮説 [Hearst 96] の下で従来手法に比べ高い性能を示すとされているが，この仮説は，ユーザは適合文書密度の最も高いクラスタのみを調べるという強い過程に基づいており，文書集合が持つ様々な話題を抽出するというクラスタリングの本来の役割はあまり考慮されていない（例えば，漠然とした検索要求を持ち，様々な話題の中から目的情報を絞り込みたいユーザには明らかに適さない仮定であるといえる）．我々がいくつかの予備実験を行った結果，STC アルゴリズムの話題抽出能力（トピック分離性能と呼ぶ）は極めて低いことが分かった．本研究は，この STC の欠点であるトピック分離性能を上げるための方法について，ベースクラスタを形成するフレーズの選択，ベースフレーズ選択後の統合という 2 つの観点からの改善する方法を提案し，評価実験を行うことによって改善された結果を検証する．

2. STC の概要

STC アルゴリズムは次の 3 つのステップからなる．

- 文書のクリーニング.
- ベースクラスタ集合の生成.
- クラスタの統合.

クリーニングに関しては，不要語の削除，語幹処理という基本的処理を行うだけなので，以下では残りの 2 つに関して簡単に説明する．

連絡先: 岡部正幸，豊橋技術科学大学マルチメディアセンター，
〒 441-8580 愛知県豊橋市天伯町雲雀ヶ丘 1-1，
Tel&Fax 0532-44-6639，okabe@mc.tut.ac.jp

2.1 ベースクラスタの生成

STC アルゴリズムは，接尾辞木（suffix tree）と呼ばれるデータ構造を利用してベースクラスタ（初期クラスタ集合）を生成する．接尾辞木とは，文書集合中の各文から接尾辞を全て抽出し，コンパクトなトライ構造としてまとめたものである [Larsson 96]．接尾辞木を生成することによって，文書集合中に現れる頻度の高いフレーズ（単語列）を効率よく見つけることができるという利点がある．

接尾辞木の中に形成されるノードは，文書集合中に現れるフレーズと対応する．このフレーズは接尾辞そのもの，または複数の接尾辞が共通して持つ接頭辞（prefix）のどちらかである．また，各ノードは対応するフレーズが出現する文書の ID 集合を持つ．STC では，これらノード集合（フレーズ集合）からベースクラスタを選び出す．各ノードには次式によるスコアが計算され，このスコアの高い n 個のノードをベースクラスタとする．

$$score(ph) = df(ph) \cdot f(len(ph)) \tag{1}$$

ここで，$df(ph)$ はフレーズ ph が持つ文書 ID の個数，$len(ph)$ は ph が持つ単語数を示す．また関数 f は ph が持つ単語数によって重要度を決定する関数であり，単語数を x とすると以下の式で示される．

$$f(x) = \begin{cases} x & (1 \leq x \leq 6) \\ 6 & (x > 6) \end{cases} \tag{2}$$

2.2 クラスタの統合

ベースクラスタはそれぞれ，フレーズとそのフレーズを含む文書集合を要素として持つ．STC では，この文書集合を基に 2 つのベースクラスタを統合するか否かを決める．2 つのベースクラスタ B_m と B_n があった場合，以下に示すように，共有文書が 2 つのクラスタに占める割合を計算し，両者が共に 0.5 以上であれば統合すると判定する．

$$\frac{|B_m \cap B_n|}{B_m} > 0.5 \text{ かつ } \frac{|B_m \cap B_n|}{B_n} > 0.5 \tag{3}$$

全ての組合せについて (3) 式の判定を行い，条件を満たすクラスタ同士は全て統合することにより最終的なクラスタ集合を形成する．

3. STC アルゴリズムの改善

STC は高速なアルゴリズムである反面，大きなクラスタを作りやすく，トピックの分離性能が悪い．この欠点を改善するための方法を以下に示す．

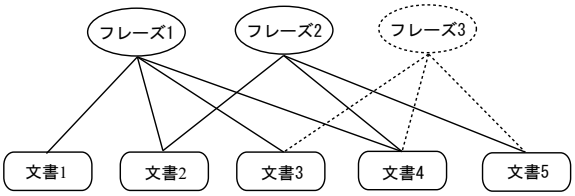

図 1: ベースクラスタの選択

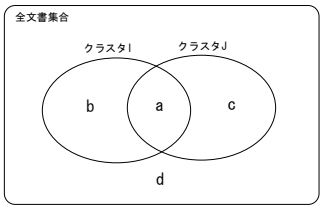

図 2: クラスタの類似度計算

3.1 情報量規準によるベースクラスタの選択

ベースクラスタ選択時の基準として重要な点は，トピック純度*1が高く，なるべく多くの文書に出現することである．STC では高速性を追求しているためスコア計算は極めて単純なものになっている．加えて，生成するベースクラスタの個数についても規準がなくアドホックに決定しなければならない（Zamir らの実験ではは n=500 としている）．このため，選択されたベースクラスタの中にトピック純度の低いものが混ざる確率が高く，文書数の大きなクラスタを作ってしまう原因となっている．

我々は，トピック純度が高くかつ多くの文書に出現する単語列を選ぶために，式 (1) と (2) を以下のように変更した．

$$score(ph) = mutual(ph) \cdot f(len(ph)) \quad (4)$$

$$mutual(ph) = \sum_{d_i \in D} p(ph, d_i) \cdot log \frac{p(ph, d_i)}{p(ph) \cdot p(d_i)} \quad (5)$$

$$f(x) = \begin{cases} 1 & (2 \leq x \leq 5) \\ 0 & (\text{それ以外}) \end{cases} \quad (6)$$

ここで，$mutual(ph)$ はフレーズ w が持つ相互情報量の値である．また，d_i は全文書集合 D の各要素，$p(ph, d_i)$ は d_i においてフレーズ ph が出現する確率，$p(ph)$，$p(d)$ はそれぞれ ph，d_i が出現する確率である．

また必要のないクラスタを排除することが重要となるため，以下のようにベースクラスタを選択する．

1. (4) 式のスコアに基づき各フレーズをソートする．
2. スコアの高いものから，もしそれ以前に選択されたフレーズが出現しない文書が少なくとも 1 つあればベースクラスタに追加する．そうでなければ，次のフレーズへ．例えば，図 1 ではフレーズ 1,2 により既に全文書がカバーされているので，フレーズ 3 は必要ない．
3. 全ての文書がベースクラスタの内の少なくとも 1 つに属することが確認されれば終了．

以上の操作を行うことによりベースクラスタ数を大幅に減らすことができる．

3.2 κ 統計量を用いた統合判定

STC では 2 章で説明した判定基準を満たす 2 つのベースクラスタは全て結合される．そのため最終的なクラスタ数をどの程度に集約させるかを調整するのが非常に困難である．また，判定基準を満たすクラスタを全て同時に統合するため，形成されるクラスタのトピック純度は一般に非常に低い．

*1 ここでは，クラスタ内において 1 つのトピックの占有率が高い程，純度が高いという意味で用いている

そこで我々は，閾値を使った統合判定ではなく，2 つのベースクラスタ間の類似度を計算する尺度を導入し，この類似度が最も高いクラスタを結合するたびに新たなクラスタと既存クラスタとの類似度を再計算する累積的階層化クラスタリングのアプローチを取る．一般にこのアプローチをとった場合，計算量が膨大なため検索結果のクラスタリングには不向きと考えられているが，先に述べたように我々の手法では，ベースクラスタの数は低く抑えられるので，逐次的な再計算に伴う計算量も低く抑えられる．

類似度を計る尺度としては，κ 統計量を用いる．κ 統計量は，一般に 2 人の判定者間の一致度を測る尺度として用いられるが，市瀬らは Web カテゴリ間の概念性の一致度を測る尺度として κ 統計量を用い，良好な結果を得ている [市瀬 02]．本研究でも同様の考え方によりベースクラスタ及び統合過程で生成される中間クラスタ間の類似尺度として κ 統計量を用いる．2 つのクラスタ I と J（ベースクラスタまたは中間クラスタ）があるとして，a をクラスタ I とクラスタ J が共通して持つ文書数，b を I がもち，かつ J がもたない文書数，c を J がもち，かつ I がもたない文書数，d を I と J が共にもたない文書数とした場合（図 2 参照），κ 統計量は以下で示される．

$$\kappa = \frac{P - P'}{1 - P'} \quad (7)$$

$$P = \frac{a + b}{a + b + c + d} \quad (8)$$

$$P' = \frac{(a + b) * (a + c) + (c + d) * (b + d)}{(a + b + c + d)^2} \quad (9)$$

通常の手続きでは (7) 式によって計算された値をもとに仮説検定を行うが，本研究では単に (7) 式の値を 2 つのクラスタ間の類似度として用いる．

4. 評価実験

3 章で説明した改善を施したアルゴリズム（κ-STC アルゴリズムと呼ぶ）の性能を調べるため，2 つの実験を行った．

4.1 実験 1：トピック分離性能の評価

実験 1 では改善によりトピック分離性能がどのくらい改善されたかを調べる．

4.2 実験方法

文書データとして TREC5[Voorhees 96] で使用された英文記事 (Foreign Broad Information Service) を用いた．TREC5 ではコンテストに使用された検索課題と，各検索課題と各文書の間の適合ラベルデータが公開されている．本実験では，1 検索課題を 1 トピックとみなし，ラベルとの照合によりトピック分離性能を評価した．また，各文書データは予め不要語と語幹処理を行っている．

表 1: 各トピックの適合文書数

トピック番号	1	5	23	24	44	54	58	77	78	114	125	126	154	173	185	192	184
適合文書数	30	19	7	38	10	48	45	14	37	42	6	18	22	15	14	10	8

各トピックの中から適合文書を 5 つ以上 50 以下含むトピックを選んで使用した．トピック番号と各トピックが持つ適合文書数を表 1 に示す．実験では，目標クラスタ数が 5, 10, 15 の場合の評価を行った（従ってこれは目標クラスタ数を予め与えてあるという条件の下での実験である）．各クラスタ数につき，表 1 からランダムに 20 組の組合せを選び，その平均で評価した．

クラスタリングの評価方法については議論があるが，本実験ではトピック分離性能を評価する値として良く用いられている相互情報量により評価した．

相互情報量

クラスタ集合を C，正解クラスタ集合[*2]を T とすると相互情報量は以下の式で計算される．

$$MI(C,T) = \sum_{c_i \in C} \sum_{t_j \in T} p(c_i, t_j) log_2 \frac{p(c_i, t_j)}{p(c_i)p(t_j)} \qquad (10)$$

この値はよく正規化して用いられているので本実験でも以下のような正規化を行った．

$$MI_{norm}(C,T) = \frac{MI(C,T)}{max(H(C), H(T))} \qquad (11)$$

$$H(X) = \sum_{x_i \in X} p(x_i) log_2 p(x_i) \qquad (12)$$

比較手法

また比較手法として，Zamir らが用いたものと同じく，Backshot，Fractionation[Cutting 92]，GAHC(Group-averaged Agglomarative Hierarchical Clustering) の各手法とオリジナルの STC を比較した．オリジナルの STC での選択フレーズ数は Zamir らが用いていた数と同じ ($n = 500$) とした．Backshot, Fractionation, GAHC の手法の各単語の重みはクラスタリング対象文書のみを用いて *tf-idf* により計算した．STC と κ-STC は生成するクラスタ間ではいくつかの文書が共有されるが，評価を行うため，重複している文書は，最も多くのフレーズに支持されているクラスタに割り当てた．

4.3 実験結果

表 2 に実験 1 の結果を示す．目標クラスタ数 (5,10,15) 別に各 20 組の平均と全て組（60 組）の平均値が示されている．オリジナルの STC は我々が指摘したように，大きなクラスタを作ってしまうため，非常に悪い結果になっている．3 手法の中では，Backshot が最も良い結果となっている．GAHC と Fractionation は階層化クラスタリングがベースとなっているが，Bachshot の結果と比べると比較的大きく偏ったクラスタを形成しており，オリジナル STC と同じく性能悪化の原因となっている．κ-STC は Backshot と同じく偏ったクラスタを形成することなく良好な性能を示している．他のクラスタでは 5 クラスタの場合の性能が良くないが，κ-STC はどのクラスタ数においても他手法を上回り，安定した性能を示している．

∗2 各文書の正解トピックラベルを用いてクラスタリングを行った集合のこと

表 2: 実験 1 の結果

	STC	GAHC	Back	Frac	κ-STC
5 クラスタ	0.06	0.10	0.31	0.25	0.51
10 クラスタ	0.09	0.18	0.46	0.36	0.53
15 クラスタ	0.12	0.17	0.45	0.40	0.47
すべて	0.09	0.15	0.43	0.34	0.51

表 3: 実験 2 の結果

	STC	GAHC	Back	Frac	κ-STC
matrix	0.16	0.18	0.27	0.24	0.40
apple	0.13	0.26	0.22	0.17	0.20
windows	0.10	0.24	0.17	0.17	0.16
science	0.09	0.16	0.23	0.21	0.33
robot	0.22	0.31	0.25	0.27	0.31
mp3	0.09	0.08	0.11	0.18	0.21
bush	0.14	0.24	0.26	0.21	0.31
time	0.20	0.26	0.40	0.34	0.45
madonna	0.13	0.22	0.30	0.23	0.28
football	0.06	0.11	0.07	0.07	0.12
全平均	0.13	0.21	0.23	0.23	0.28

ちなみに，Backshot と Fractionation の 2 手法は同じ文書数でもクラスタ数が増加すると計算量が大きく増加する．κ-STC は特徴語選択の調整は必要なく，計算量は文書数のみに依存する．

4.4 実験 2：Web 検索における有効性

実験 1 と同様の実験を Web 検索エンジン Google を用いて行った．検索課題として 10 個のクエリ（'matrix','apple','windows','science','robot','mp3','bush','time', 'madonna','football'）を Google に入力した．この実験では Web ページ内のテキスト情報を用いるのではなく，GoogleAPI[*3]を使って得られるタイトル，抜粋文 (snippet)，要約文 (summary) の情報のみを使ってクラスタリングを行った．また正解ラベルとして GoogleAPI を使って得られるディレクトリ情報（カテゴリ情報）の第 1 階層を使用した．例えば，ディレクトリ情報が，'Top/Computers/Systems/Apple/Macintosh/' であった場合，'Computers' を正解ラベルとして用いる．ディレクトリ情報はすべてのページ割り振られているわけではないので，ヒットリストからディレクトリ情報をもつ上位 100 個の URL をクラスタリング対象とした．

表 3 に実験結果を示す．各手法の，各クエリにおける相互情報量の値が示されている．ほとんどのクエリにおいて κ-STC が高い性能を示しており，Web 検索において有効であることが示されている．

∗3 http://www.google.com/apis/

5. まとめ

本研究では，高速クラスタリングアルゴリズム STC のトピック分離性能を改善する手法について述べた．提案手法によりトピック分離性能が大幅に向上したことを確認した．また，提案手法は従来のクラスタリング手法に比べて高い性能を示し，目標クラスタ数に依らない安定した性能を示す．今後の課題として，ベースクラスタ選択基準の改良が挙げられる．また，クラスタ数のバリエーションを変えた場合の性能と計算量の比較や Web 検索における比較をより詳細に行う予定である．

参考文献

[Zamir 98] Zamir, O. and Etzioni, O.: Web Document Clustering: A Feasibility Demonstration, in Proc. of the 21st International ACM SIGIR Conference, pp.46-54 (1998)

[Hearst 96] Hearst, M. A. and Pedersen, J. O.:Reexamining the cluster hypothesis: Scatter/Gather on retrieval results, In Proceedings of the 19th International ACM SIGIR Conference, pp.76-84 (1996)

[Larsson 96] Larsson, N. J.: Extended Application of Suffix Trees to Data Compression, Data Compression Conference, pp.190-199 (1996)

[市瀬 02] 市瀬 龍太郎，武田 英明，本位田 真一: 階層知識間の調整規則の学習，人工知能学会誌，Vol.17, No.3, pp.230-238 (2002)

[Voorhees 96] Voorhees, E. M. and Harman, D. K.:NIST Special Publication 500-238: The Fifth Text REtrieval Conference (TREC-5) (1996)

[Cutting 92] Cutting, D. R. et al.:Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections, In Proceedings of the 15th International ACM SIGIR Conference, pp.318-329 (1992)